aws

Guide de l’utilisateur

AWS Service de Migration de Base de Données

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Service de Migration de Base de Données: Guide de l’utilisateur

Copyright © 2024 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

Qu'est-ce que AWS Database Migration Service ? ... 1
Tâches de migration assurées par AWS DMS ... 2
Comment AWS DMS fonctionne ... 4
Vue de haut niveau de AWS DMS ... 4
Composants ... 6
Sources ... 13
Sources pour la migration des données ... 14
Sources pour DMS Fleet Advisor ... 17
Sources pour la conversion de schéma DMS ... 17
Sources pour les migrations de données homogènes DMS ... 18
Cibles ... 18
Cibles pour la migration des données ... 19
Cibles pour DMS Fleet Advisor ... 21
Cibles pour la conversion de schéma DMS ... 22
Cibles pour les migrations de données homogènes DMS ... 22
Amazon Resource Names ... 23
Avec d'autres AWS services ... 25
Support pour AWS CloudFormation ... 26
Démarrer ... 27
Configuration ... 28
S'inscrire à un Compte AWS ... 28
Création d'un utilisateur administratif ... 28
Prérequis ... 29
Création d'un VPC ... 30
Création de groupes de paramètres Amazon RDS ... 32
Création de la base de données Amazon RDS source ... 33
Création de la base de données Amazon RDS cible ... 35
Création d'un client Amazon EC2 ... 36
Remplissage de la base de données source ... 37
Migration du schéma ... 39
Réplication ... 41
Étape 1 : Créer une instance de réplication ... 41
Étape 2 : Spécifier les points de terminaison sources et cibles ... 44
Étape 3 : Créer une tâche et migrer les données ... 45

Étape 4 : Tester la réplication 47
Étape 5 : Nettoyer les ressources AWS DMS 50
Ressources supplémentaires 51
Découverte des bases de données pour la migration 52
Régions AWS prises en charge 53
Démarrer 55
Configuration 56
Création des ressources requises 56
Création d'utilisateurs de base de données 66
Collecteurs de données 73
Autorisations 74
Création d'un collecteur de données 75
Installation d'un collecteur de données 77
Découverte des serveurs de base de données et de système d'exploitation 81
Gestion des objets surveillés 84
Utilisation de SSL 87
Collecte de données 88
Résolution des problèmes 93
Inventory 96
Utilisation de l'inventaire des bases de données 97
Utilisation de l'inventaire des schémas 98
Recommandations cibles 100
Instances cibles 101
Comment DMS Fleet Advisor détermine-t-il les spécifications cibles ? 101
Génération de recommandations cibles 102
Détails des recommandations 105
Exportation des recommandations cibles 106
Limites de migration 108
Résolution des problèmes 129
Limites 130
Conversion de schémas de base de données 132
Soutenu Régions AWS 133
Fonctionnalités 134
Limites 135
Premiers pas 136
Prérequis 137

Étape 1 : Créer un profil d'instance 142
Étape 2 : Configurer des fournisseurs de données 143
Étape 3 : Créer un projet de migration 144
Étape 4 : Créer un rapport d'évaluation 145
Étape 5 : Convertir le code source 146
Étape 6 : Appliquer le code converti 146
Étape 7 : Nettoyer et résoudre les problèmes 147
Configuration d'un réseau 148
Configuration de VPC unique 148
Configuration de plusieurs VPC 149
Utilisation d'AWS Direct Connect ou d'un VPN 149
Utilisation d'une connexion Internet 150
Utilisation d'un environnement sans passerelle Internet 150
Création de fournisseurs de données sources 151
Utilisation de SQL Server comme source 152
Utilisation d'Oracle comme source 153
Utilisation d'Oracle Data Warehouse en tant que source 154
Utilisation de PostgreSQL comme source 157
Utilisation de MySQL comme source 158
Création de fournisseurs de données cibles 159
Utilisation de MySQL en tant que cible 159
Utilisation de PostgreSQL comme cible 161
Utilisation d'Amazon Redshift en tant que cible 162
Gestion de projets de migration 163
Spécification des paramètres de projet de migration 163
Rapports d'évaluation de migration de base de données 165
Création d'un rapport d'évaluation 165
Affichage de votre rapport d'évaluation 166
Enregistrement de rapports d'évaluation 167
Conversion du schéma 169
Configuration des règles de transformation 170
Conversion de votre schéma de base de données 173
Spécification des paramètres de conversion de schéma 176
Actualisation de vos schémas de base de données 182
Enregistrement et application de votre schéma 183
Utilisation de packs d'extension 185

Migrations de données homogènes 187
Régions AWS prises en charge 188
Fonctionnalités 189
Limites 190
Présentation 190
Configuration 191
Création de ressources IAM 192
Configuration d'un réseau 196
Création de fournisseurs de données sources 201
Utilisation de MySQL ou MariaDB en tant que source 201
Utilisation de PostgreSQL comme source 206
Utiliser MongoDB ou Amazon DocumentDB comme source 209
Création de fournisseurs de données cibles 213
Utilisation de MySQL ou MariaDB en tant que cible 214
Utilisation de PostgreSQL comme cible 215
Utilisation d'Amazon DocumentDB en tant que cible 217
Migration de données 218
Création d'une migration de données 218
Règles de sélection 221
Gestion des migrations de données 224
Surveillance des migrations de données 225
Statuts de migration 227
Migration de données à partir de MySQL 229
Migration de données à partir de PostgreSQL 230
Migration de données depuis MongoDB 232
Résolution des problèmes 234
Créer une migration de données 234
Démarrer une migration de données 234
Problèmes de connexion 235
Les vues sont migrées en tant que tables dans PostgreSQL 236
Utilisation de projets de migration 237
Création d'un groupe de sous-réseaux 238
Création de profils d'instance 239
Création de fournisseurs de données 240
Création de projets de migration 243
Gestion des projets de migration 244

Bonnes pratiques 246
Planification de la migration pour AWS Database Migration Service 246
Conversion du schéma 248
Examen de la documentation d'AWS DMS 248
Exécution d'une preuve de concept 249
Amélioration des performances 249
Utilisation de votre propre serveur de noms sur site 255
Utilisation d'Amazon Route 53 Resolver avec AWS DMS 256
Migration des objets binaires volumineux (Large Binary Object, LOB) 257
Utilisation du mode LOB limité 257
Amélioration des performances des objets LOB 259
Amélioration des performances en cas de migration de tables volumineuses à l'aide du filtrage des lignes 262
la réplication continue. 262
Réduction de la charge sur votre base de données source 263
Réduction des goulots d'étranglement sur la base de données cible 264
Utilisation de la validation des données 264
Surveillance des métriques 265
Événements 265
Utilisation du journal des tâches 266
Résolution des problèmes de réplication à l'aide du voyage dans le temps 266
Modification de l'utilisateur et du schéma pour une cible Oracle 267
Modification des espaces de table d'index et de table pour une cible Oracle 267
Mise à niveau d'une instance de réplication 269
Comprendre les coûts de migration 269
Travailler avec AWS DMS Serverless 270
Composants de DMS sans serveur 271
Versions de moteur prises en charge 274
Création d'une réplication sans serveur 275
Modification des AWS DMS réplications sans serveur 277
Configuration de calcul 281
Comprendre la mise à l'échelle automatique en mode sans serveur AWS DMS 283
Surveillance des AWS DMS réplications sans serveur 284
Limitations sans serveur 289
Utilisation d'instances de réplication 291
Choix des types d'instances de réplication 296

Choix de la classe d'instances à utiliser 301
Instances extensibles en mode illimité 302
Dimensionnement d'une instance de réplication 303
Facteurs à prendre en compte 304
Problèmes courants 305
Bonnes pratiques 306
Versions du moteur de réplication 307
Mise à niveau de la version du moteur à l'aide de la console 307
Mise à niveau de la version du moteur à l'aide du AWS CLI 308
Instances de réplication publiques et privées 309
Adressage IP et types de réseau 309
Configuration d'un réseau pour une instance de réplication 311
Configurations réseau pour la migration de base de données 312
Créer groupe de sous-réseaux de réplication 321
Résolution des points de terminaison de domaine à l'aide de DNS 323
Définition d'une clé de chiffrement 323
Création d'une instance de réplication 324
Modification d'une instance de réplication 331
redémarrage d'une instance de réplication. 337
Supprimez une instance de réplication. 340
Fenêtre de maintenance DMS 342
Effet de la maintenance sur les tâches de migration existantes 342
Modification de la configuration de la fenêtre de maintenance 343
Points de terminaison 345
Création de points de terminaison source et cible 345
Sources pour la migration des données 351
Utilisation d'Oracle comme source 352
Utilisation de SQL Server comme source 425
Utilisation d'Azure SQL Database comme source 457
Utilisation de Azure SQL Managed Instance en tant que source 457
Utilisation d'Azure Database pour PostgreSQL en tant que source 458
Utilisation d'Azure Database pour MySQL en tant que source 459
Utilisation d'OCI MySQL Heatwave en tant que source 460
Utilisation de Google Cloud pour MySQL en tant que source 461
Utilisation de Google Cloud pour PostgreSQL en tant que source 461
Utilisation de PostgreSQL comme source 463

Utilisation de MySQL comme source ... 502
Utilisation de SAP ASE comme source ... 516
Utilisation de MongoDB comme source ... 525
Utilisation d'Amazon DocumentDB en tant que source ... 544
Utilisation d'Amazon S3 en tant que source ... 562
Utilisation d'IBM Db2 LUW comme source ... 575
Utilisation d'IBM Db2 for z/OS en tant que source ... 583
Cibles pour la migration des données ... 624
Utilisation d'Oracle comme cible ... 626
Utilisation de SQL Server comme cible ... 637
Utilisation de PostgreSQL comme cible ... 644
Utilisation de MySQL en tant que cible ... 657
Utilisation d'Amazon Redshift en tant que cible ... 666
Utilisation de SAP ASE comme cible ... 693
Utilisation d'Amazon S3 en tant que cible ... 696
Utilisation d'Amazon DynamoDB en tant que cible ... 750
Utilisation d'Amazon Kinesis Data Streams en tant que cible ... 772
Utilisation d'Apache Kafka comme cible ... 792
Utilisation d'OpenSearch comme cible ... 820
Utilisation d'Amazon DocumentDB en tant que cible ... 827
Utilisation d'Amazon Neptune en tant que cible ... 841
Utilisation de Redis en tant que cible ... 858
Utilisation de Babelfish en tant que cible ... 866
Utilisation d'Amazon Timestream comme cible ... 875
Utilisation de Db2 en tant que cible ... 887
Points de terminaison de VPC pour la migration des données ... 888
Qui est affecté lors d'une migration vers AWS DMS version 3.4.7 ou ultérieure ? ... 889
Qui n'est pas affecté lors d'une migration vers AWS DMS version 3.4.7 ou ultérieure ? ... 889
Préparation d'une migration vers AWS DMS version 3.4.7 ou ultérieure ... 890
Instructions DDL prises en charge ... 891
Tâches ... 893
Création d'une tâche ... 898
Paramètres de tâche ... 907
Définition de la prise en charge LOB ... 963
Création de plusieurs tâches ... 965
Tâches de réplication continue ... 965

Réplication à partir d'un point de départ CDC ... 967
Effectuer une réplication bidirectionnelle ... 973
Modification d'une tâche ... 977
Déplacement d'une tâche ... 977
Rechargement de tables pendant une tâche ... 978
AWS Management Console ... 979
Mappage de table ... 980
Spécification des règles de sélection de table et de transformation à partir de la console ... 981
Spécification des règles de sélection de table et de transformation à l'aide de JSON ... 985
Règles et actions de sélection ... 987
Caractères génériques dans le mappage de table ... 994
Règles et actions de transformation ... 995
Utilisation d'expressions de règle de transformation pour définir le contenu d'une colonne . 1019
Règles des paramètres de table et de collection et opérations ... 1034
Utilisation de filtres de source ... 1067
Application de filtres ... 1068
Filtrage par heure et date ... 1074
Activation et utilisation des évaluations de prémigration ... 1075
Prérequis ... 1077
Spécification, démarrage et affichage des exécutions d'évaluation ... 1079
Évaluations individuelles ... 1084
Démarrage et affichage des évaluations du type de données ... 1106
Exécutions d'évaluation du dépannage ... 1111
Spécification de données supplémentaires ... 1112
Surveillance des tâches ... 1113
État de la tâche ... 1115
État d'une table pendant des tâches ... 1118
Surveillance des tâches de réplication à l'aide d'Amazon CloudWatch ... 1120
Métriques AWS Database Migration Service ... 1121
Métriques des instances de réplication ... 1125
Métriques de tâches de réplication ... 1127
Affichage et gestion des journaux AWS DMS ... 1131
Journalisation des appels d'API AWS DMS avec AWS CloudTrail ... 1133
AWS DMS Informations dans CloudTrail ... 1133
Présentation des AWS DMS entrées des fichiers journaux ... 1134
Journalisation du contexte ... 1138

Types d'objets 1138
Exemples de journalisation 1140
Limites 1141
Utilisation d'événements EventBridge 1142
Utilisation des règles d'événement Amazon EventBridge pour AWS DMS 1143
Catégories d'événements et messages d'événements AWS DMS 1144
Messages d'événements ReplicationInstance 1144
Messages d'événements ReplicationTask 1149
Messages d'événements Replication 1151
Utilisation des évènements Amazon SNS 1153
Transfert des abonnements aux événements vers Amazon EventBridge 1153
Utilisation des événements et des notifications Amazon SNS 1154
Catégories d'événements et messages d'événements AWS DMS pour les notifications SNS . 1156
Abonnement aux notifications d'événement AWS DMS à l'aide de SNS 1160
Utilisation de AWS Management Console 1160
Validation de la stratégie d'accès de votre rubrique SNS 1163
Validation des données 1165
Statistiques des tâches de réplication 1166
Statistiques des tâches de réplication avec Amazon CloudWatch 1169
Revalidation de tables pendant une tâche 1170
AWS Management Console 1170
Utilisation de l'éditeur JSON pour modifier les règles de validation 1171
Tâches de validation uniquement 1172
Validation du chargement complet uniquement 1172
Validation CDC uniquement 1173
Cas d'utilisation relatifs à la validation uniquement 1173
Résolution des problèmes 1174
Performances de validation Redshift 1176
Limites 1177
Validation S3 1178
Prérequis 1179
Autorisations 1180
Limites 1181
Tâches de validation uniquement 1182
Étiquetage des ressources 1184
API 1186

Sécurité 1188
Protection des données 1191
Chiffrement des données 1191
Confidentialité du trafic inter-réseau 1192
Protection des données dans DMS Fleet Advisor 1193
Gestion des identités et des accès 1194
Public ciblé 1194
Authentification par des identités 1195
Gestion des accès à l'aide de politiques 1199
Comment AWS Database Migration Service fonctionne avec IAM 1201
Exemples de politiques basées sur l'identité 1209
Exemples de stratégies basées sur les ressources 1218
Utilisation de secrets pour accéder aux ressources 1223
Utilisation des rôles liés aux services 1233
Résolution des problèmes 1240
Autorisations IAM nécessaires 1243
Rôles IAM pour l'interface de ligne de commande et l'API 1248
Prévention du problème de l'adjoint confus entre services 1254
Stratégies gérées par AWS 1257
Validation de la conformité 1264
Résilience 1266
Sécurité de l'infrastructure 1267
Contrôle précis des accès 1271
Utilisation des noms de ressources pour contrôler l'accès 1271
Utilisation de balises pour contrôler l'accès 1274
Définition d'une clé de chiffrement 1282
Sécurité du réseau 1285
Utilisation de SSL 1287
Limitations sur l'utilisation de SSL avec AWS DMS 1289
Gestion de certificats 1290
Activation de SSL pour un point de terminaison compatible MySQL, PostgreSQL ou SQL Server 1290
Modification du mot de passe de base de données 1293
Limites 1294
Quotas de ressources pour AWS Database Migration Service 1294
Comprendre la limitation des demandes d'API 1296

Résolution des problèmes et assistance au diagnostic 1297
Les tâches de migration s'exécutent lentement 1298
La barre de statut des tâches ne bouge pas 1299
La tâche est terminée mais rien n'a été migré 1299
Des clés étrangères et des index secondaires sont manquants 1299
AWS DMSne crée pas de CloudWatch journaux 1300
Des problèmes se produisent lors de la connexion à Amazon RDS 1300
Message d'erreur : Incorrect thread connection string: incorrect thread value 0 1301
Des problèmes de réseau surviennent 1301
La capture des données modifiées est bloquée après le chargement complet 1302
Erreurs de violation de clé primaire lors du redémarrage d'une tâche 1303
Échec du chargement initial d'un schéma 1303
Échec des tâches avec une erreur inconnue 1303
Le redémarrage d'une tâche charge les tables dès le début 1303
Le nombre de tables par tâche pose problème 1304
Les tâches échouent quand une clé primaire est créée sur une colonne LOB 1304
Des doublons d'enregistrements apparaissent sur une table cible sans clé primaire 1304
Échec des points de terminaison sources dans la plage IP réservée 1304
Les horodatages sont brouillés dans les requêtes Amazon Athena 1305
Résolution de problèmes avec Oracle 1305
Extraire les données à partir de vues 1306
Migration des LOB à partir d'Oracle 12c 1306
Basculer entre Oracle LogMiner et Binary Reader 1306
Erreur : CDC Oracle arrêtée 122301 nombre maximal de nouvelles tentatives de CDC Oracle dépassé. 1307
Ajout automatique d'une journalisation supplémentaire à un point de terminaison source Oracle 1307
Les modifications de LOB ne sont pas capturées 1308
Erreur : ORA-12899 : valeur trop grande pour la colonne *nom de la colonne* 1308
Mauvaise interprétation du type de données NUMBER 1309
Enregistrements manquants pendant le chargement complet 1309
Erreur de table 1309
Erreur : impossible de récupérer les identifiants de destination des journaux Redo archivés par Oracle 1310
Évaluation des performances de lecture des journaux redo ou d'archivage Oracle 1310
Résolution des problèmes liés à MySQL 1312

Échec de la tâche CDC pour le point de terminaison d'instance DB Amazon RDS car la journalisation binaire est désactivée 1313
Les connexions à une instance MySQL cible sont déconnectées durant une tâche 1313
Ajout de la validation automatique à un point de terminaison compatible MySQL 1314
Désactiver les clés étrangères sur un point de terminaison cible compatible MySQL 1315
Caractères remplacés par un point d'interrogation 1315
Entrées de journal « Événement incorrect » 1315
Capture de données modifiées avec MySQL 5.5 1316
Augmentation de la durée de conservation des journaux binaires pour les instances de base de données Amazon RDS 1316
Message du journal : quelques modifications de la base de données source n'ont eu aucun impact lorsqu'elles ont été appliquées à la base de données cible. 1316
Erreur : Identificateur trop long 1316
Erreur : un jeu de caractères non pris en charge entraîne l'échec de la conversion des données de champ 1317
Erreur : page de codes 1252 à UTF8 [120112] Échec de la conversion des données d'un champ 1317
Non-migration des index, des clés étrangères ou des mises à jour ou suppressions en cascade 1318
Résolution des problèmes liés à PostgreSQL 1320
Types de données JSON tronqués 1321
Migration incorrecte de colonnes d'un type de données défini par l'utilisateur 1321
Erreur : Aucun schéma sélectionné dans lequel effectuer la création 1322
Les suppressions et les mises à jour dans une table ne sont pas répliquées via la CDC 1322
Les instructions de troncation ne sont pas propagées 1322
Empêcher PostgreSQL de capturer la DDL 1322
Sélectionner le schéma où sont créés les objets de base de données pour la capture de la DDL 1323
Tables Oracle manquantes après la migration vers PostgreSQL 1323
ReplicationSlotDiskUsage augmente et restart_lsn cesse d'avancer pendant les transactions longues, telles que les charges de travail ETL 1323
Une tâche utilisant une vue comme source ne contient aucune ligne copiée 1323
Résolution des problèmes liés à Microsoft SQL Server 1324
Erreurs de capture des modifications pour une base de données SQL Server 1324
Colonnes d'identité manquantes 1325
Erreur : SQL Server ne prend pas en charge les publications 1325

Les modifications ne s'affichent pas dans votre cible 1325
Table non uniforme mappée entre les partitions 1326
Résolution des problèmes liés à Amazon Redshift 1326
Chargement dans un cluster Amazon Redshift dans une autre région AWS 1327
Erreur : la relation « awsdms_apply_exceptions » existe déjà 1327
Erreurs avec les tables dont le nom commence par « awsdms_changes » 1327
Présence de tables dans les clusters avec des noms du type dms.awsdms_changes000000000XXXX 1327
Autorisations requises pour utiliser Amazon Redshift 1328
Résolution des problèmes liés à Amazon Aurora MySQL 1328
Erreur : Champs CHARACTER SET UTF8 se terminant par une « , » entourée par des lignes « " » se terminant par « \n » 1328
Résolution des problèmes liés à SAP ASE 1329
Erreur : les colonnes LOB ont des valeurs NULL lorsque la source possède un index unique composite avec des valeurs NULL 1329
Résolution des problèmes liés à IBM Db2 1329
Erreur : Resume from timestamp is not supported Task 1329
Résolution des problèmes de latence 1330
Types de latence de CDC 1330
Causes courantes de la latence de CDC 1331
Résolution des problèmes de latence 1335
Utilisation de scripts d'assistance au diagnostic 1351
Scripts d'assistance Oracle 1353
Scripts d'assistance SQL Server 1356
Scripts d'assistance compatibles MySQL 1382
Scripts d'assistance PostgreSQL 1384
Utilisation de l'AMI d'assistance au diagnostic 1387
Lancement d'une nouvelle instance Amazon EC2 de diagnostic AWS DMS 1388
Créer un rôle IAM 1388
Exécution de tests de diagnostic 1389
Étapes suivantes 1393
ID d'AMI par région 1394
Référence 1396
Types de données AWS DMS 1397
Notes de mise à jour 1399
AWS DMS notes de mise à jour 3.5.2 1399

AWS DMS notes de mise à jour 3.5.1 1400
AWS DMS Notes de mise à jour de la version bêta 3.5.0 1412
AWS DMS 3.4.7 notes de mise à jour 1418
AWS DMS notes de mise à jour 3.4.6 1428
AWS DMS notes de mise à jour 3.4.5 1435
AWS DMS 3.4.4 notes de mise à jour 1439
AWS DMS 3.4.3 notes de mise à jour 1441
AWS DMS 3.4.2 notes de mise à jour 1444
AWS DMS 3.4.1 notes de mise à jour 1446
AWS DMS notes de mise à jour de la 3.4.0 1448
AWS DMS 3.3.4 notes de mise à jour 1450
AWS DMS 3.3.3 notes de mise à jour 1450
Historique du document 1453
Glossaire AWS 1458
..... mcdlix

Qu'est-ce que AWS Database Migration Service ?

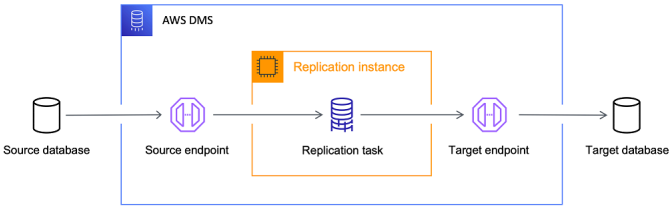
AWS Database Migration Service (AWS DMS) est un service cloud qui permet la migration des bases de données relationnelles, des entrepôts de données, des bases de données NoSQL et d'autres types de magasins de données. Vous pouvez utiliser AWS DMS pour migrer vos données dans le AWS Cloud ou entre différentes combinaisons de configurations cloud et sur site.

Avec AWS DMS, vous pouvez découvrir vos magasins de données sources, convertir vos schémas sources et migrer vos données.

- Pour découvrir votre infrastructure de données source, vous pouvez utiliser DMS Fleet Advisor. Ce service collecte les données de vos serveurs d'analytique et de base de données sur site, et dresse un inventaire des serveurs, des bases de données et des schémas que vous pouvez migrer vers le Cloud AWS.
- Pour migrer vers un autre moteur de base de données, vous pouvez utiliser la conversion de schéma DMS. Ce service évalue et convertit automatiquement les schémas sources en un nouveau moteur cible. Vous pouvez également télécharger l'AWS Schema Conversion Tool (AWS SCT) sur votre PC local pour convertir les schémas sources.
- Après avoir converti les schémas sources et appliqué le code converti à la base de données cible, vous pouvez utiliser AWS DMS pour migrer vos données. Vous pouvez effectuer des migrations uniques ou répliquer les modifications continues pour maintenir la synchronisation des sources et des cibles. Comme AWS DMS fait partie intégrante du AWS Cloud, vous bénéficiez des avantages des services AWS en termes de coûts, de rapidité de mise sur le marché, de sécurité et de flexibilité.

À un niveau élémentaire, AWS DMS est un serveur situé dans le AWS Cloud qui exécute un logiciel de réplication. Vous créez une connexion source et cible pour indiquer à AWS DMS où extraire les données et où les charger. Vous planifiez ensuite une tâche qui s'exécute sur ce serveur pour déplacer vos données. AWS DMS crée les tables et les clés primaires associées si elles n'existent pas sur la cible. Si vous le préférez, vous pouvez créer vous-même les tables cibles. Vous pouvez également utiliser AWS Schema Conversion Tool (AWS SCT) pour créer tout ou partie des tables cibles, des index, des vues, des déclencheurs, etc.

Le diagramme suivant illustre le processus de réplication AWS DMS.



Références

- Régions AWS qui prennent en charge AWS DMS : pour en savoir plus sur les régions AWS qui prennent en charge AWS DMS, consultez Utilisation d'une instance AWS DMS de réplication.
- Coût de la migration de base de données : pour en savoir plus sur le coût de la migration d'une base de données, consultez la page de tarification d'AWS Database Migration Service.
- Fonctionnalités et avantages d'AWS DMS : pour en savoir plus sur les fonctionnalités et les avantages d'AWS DMS, consultez Fonctions d'AWS Database Migration Service.
- Options de base de données disponibles : pour en savoir plus sur les différentes options de base de données disponibles sur Amazon Web Services, consultez Choisir la base de données adaptée à votre organisation.

Tâches de migration assurées par AWS DMS

AWS DMS prend à sa charge la plupart des tâches complexes ou fastidieuses qu'implique un projet de migration :

- Dans une solution classique, vous devez procéder à une analyse de la capacité, acheter du matériel et des logiciels, installer et administrer des systèmes, puis tester et déboguer l'installation. Avec AWS DMS, le déploiement, la gestion et la surveillance de l'ensemble du matériel et des logiciels nécessaires à la migration sont automatiquement gérés. Votre migration peut être opérationnelle en quelques minutes après avoir lancé le processus de configuration AWS DMS.
- Avec AWS DMS, vous pouvez augmenter (ou diminuer) la capacité de vos ressources de migration pour l'adapter à votre charge de travail effective. Par exemple, si vous constatez que vous avez besoin d'une capacité de stockage supplémentaire, vous pouvez facilement augmenter l'espace de stockage qui vous est alloué et redémarrer votre migration, généralement en quelques minutes.
- AWS DMS utilise un modèle de tarification à l'utilisation. Contrairement aux modèles de licence classiques qui impliquent des coûts d'achat initiaux et des frais de maintenance continus, vous ne payez que pour le temps d'utilisation des ressources AWS DMS.

- AWS DMS gère automatiquement l'ensemble de l'infrastructure qui prend en charge votre serveur de migration, notamment le matériel et les logiciels, l'application de correctifs logiciels et le signalement des erreurs.
- AWS DMS assure un basculement automatique. Si votre serveur de réplication principal connaît une défaillance pour une raison quelconque, un serveur de réplication de secours peut prendre le relais avec peu ou pas d'interruption de service.
- AWS DMS Fleet Advisor dresse automatiquement l'inventaire de votre infrastructure de données. Il crée des rapports qui vous aideront à identifier les candidats à la migration et à planifier votre migration.
- La conversion de schéma AWS DMS évalue automatiquement la complexité de votre migration pour votre fournisseur de données source. Il convertit également les schémas de base de données et les objets de code dans un format compatible avec la base de données cible, puis applique le code converti.
- AWS DMS peut vous aider à passer à un moteur de base de données moderne, peut-être plus rentable que celui que vous exécutez actuellement. Par exemple, AWS DMS peut vous aider à tirer parti des services de base de données gérés fournis par Amazon Relational Database Service (Amazon RDS) ou Amazon Aurora. Il peut également vous aider à adopter le service d'entrepôt de données géré proposé par Amazon Redshift, des plateformes NoSQL comme Amazon DynamoDB ou des plateformes de stockage à faible coût comme Amazon Simple Storage Service (Amazon S3). Inversement, si vous souhaitez migrer à partir d'une ancienne infrastructure mais continuer à utiliser le même moteur de base de données, AWS DMS prend également en charge ce processus.
- AWS DMS prend en charge presque tous les moteurs SGBD les plus populaires d'aujourd'hui en tant que points de terminaison sources. Pour de plus amples informations, veuillez consulter Sources pour la migration des données.
- AWS DMS fournit une large couverture des moteurs cibles disponibles. Pour de plus amples informations, veuillez consulter Cibles pour la migration des données.
- Vous pouvez migrer de n'importe quelle source de données prise en charge vers n'importe quelle cible de données prise en charge. AWS DMS prend en charge les migrations de données entièrement hétérogènes entre les moteurs pris en charge.
- AWS DMS garantit une migration de données sécurisée. Les données au repos sont chiffrées à l'aide du chiffrement AWS Key Management Service (AWS KMS). Au cours de la migration, vous pouvez utiliser le protocole SSL (Secure Socket Layers) pour chiffrer vos données à la volée pendant qu'elles transitent de la source vers la cible.

Fonctionnement du Service AWS de Migration de Base de Données

AWS Database Migration Service (AWS DMS) est un service Web que vous pouvez utiliser pour migrer des données d'un magasin de données source vers un magasin de données cible. Ces deux magasins de données sont appelés « points de terminaison ». Vous pouvez effectuer des migrations entre des points de terminaison source et cible qui utilisent le même moteur de base de données, par exemple d'une base de données Oracle vers une base de données Oracle. Vous pouvez également procéder à une migration entre des points de terminaison source et cible qui utilisent des moteurs de base de données différents, par exemple d'une base de données Oracle vers une base de données PostgreSQL. La seule exigence d'utilisation AWS DMS est que l'un de vos points de terminaison se trouve sur un AWS service. Vous ne pouvez pas l'utiliser AWS DMS pour migrer d'une base de données locale vers une autre base de données locale.

Pour plus d'informations sur le coût de la migration d'une base de données, consultez la page de tarification AWS Database Migration Service.

Utilisez les rubriques suivantes pour mieux comprendre AWS DMS.

Rubriques

- Vue de haut niveau de AWS DMS
- Composants de AWS DMS
- Sources pour AWS DMS
- Objectifs pour AWS DMS
- Création d'un nom de ressource Amazon (ARN) pour AWS DMS
- Utilisation AWS DMS avec d'autres AWS services

Vue de haut niveau de AWS DMS

Pour effectuer une migration de base de données, il faut AWS DMS se connecter au magasin de données source, lire les données sources et formater les données pour qu'elles soient consommées par le magasin de données cible. Ensuite, il charge les données dans le magasin de données cible. La majeure partie de ce traitement se passe dans la mémoire, même si les transactions importantes peuvent avoir besoin d'une mise en mémoire tampon sur le disque. Les transactions mises en cache et les fichiers journaux sont également écrits sur le disque.

À un niveau élevé, lors de l'utilisation, AWS DMS vous devez effectuer les opérations suivantes :

- Découvrir les bases de données de votre environnement réseau qui sont de bons candidats à la migration.
- Convertir automatiquement les schémas de la base de données source et la plupart des objets de code de base de données dans un format compatible avec la base de données cible.
- Créer un serveur de réplication.
- Créer les points de terminaison source et cible ayant les informations de connexion sur vos magasins de données.
- Créer une ou plusieurs tâches de migration pour migrer les données entre les magasins de données source et cible.

Une tâche se compose de trois phases principales :

- Migration des données existantes (chargement complet)
- l'application des modifications mises en cache ;
- Réplication continue (capture des données de modification)

Lors d'une migration à chargement complet, lorsque les données existantes de la source sont déplacées vers la cible, AWS DMS charge les données des tables du magasin de données source vers les tables du magasin de données cible. Lorsque le chargement complet est en cours, toutes les modifications apportées aux tables en cours de chargement sont mises en cache sur le serveur de réplication ; ce sont les modifications mises en cache. Il est important de noter que AWS DMS cela ne capture pas les modifications apportées à une table donnée tant que le chargement complet de cette table n'est pas commencé. En d'autres termes, le point de démarrage de la capture des modifications varie d'une table à une autre.

Lorsque le chargement complet d'une table donnée est terminé, commence AWS DMS immédiatement à appliquer les modifications mises en cache pour cette table. Une fois la table chargée et les modifications mises en cache appliquées, AWS DMS commence à collecter les modifications sous forme de transactions pour la phase de réplication en cours. Si les tables d’une transaction ne sont pas encore complètement chargées, les modifications sont stockées localement sur l’instance de réplication. Une fois toutes AWS DMS les modifications mises en cache appliquées, les tables sont cohérentes sur le plan des transactions. À ce stade, AWS DMS passe à la phase de réplication en cours, en appliquant les modifications sous forme de transactions.

Au début de la phase de réplication continue, un retard de transactions entraîne généralement un décalage entre les bases de données source et cible. La migration atteint finalement un état est stable, après avoir traité ce retard de transactions. À ce stade, vous pouvez fermer vos applications, autoriser l'application des transactions restantes à la cible, relancer vos applications, qui pointent maintenant vers la base de données cible.

AWS DMS crée les objets du schéma cible nécessaires pour effectuer une migration de données. Vous pouvez AWS DMS utiliser une approche minimaliste et créer uniquement les objets nécessaires pour migrer efficacement les données. Cette approche permet de AWS DMS créer des tables, des clés primaires et, dans certains cas, des index uniques, mais elle ne crée aucun autre objet inutile pour migrer efficacement les données depuis la source.

Vous pouvez également utiliser la conversion de schéma DMS intégrée AWS DMS pour convertir automatiquement les schémas de votre base de données source et la plupart des objets de code de base de données dans un format compatible avec la base de données cible. Cette conversion inclut les tables, les vues, les procédures stockées, les fonctions, les types de données, les synonymes, etc. Tous les objets que la conversion de schéma DMS ne peut pas convertir automatiquement sont clairement marqués. Pour terminer la migration, vous pouvez convertir ces objets manuellement.

Composants de AWS DMS

Cette section décrit les composants internes AWS DMS et la manière dont ils fonctionnent ensemble pour effectuer la migration de vos données. La compréhension des composants sous-jacents d' AWS DMS peut vous aider à migrer vos données plus efficacement et à mieux saisir les tenants et les aboutissants lors de la résolution ou de l'étude de problèmes.

Une AWS DMS migration comprend cinq éléments : découverte des bases de données à migrer, conversion automatique du schéma, instance de réplication, points de terminaison source et cible, et tâche de réplication. Vous créez une AWS DMS migration en créant l'instance de réplication, les points de terminaison et les tâches nécessaires dans un Région AWS.

Découverte des bases de données

DMS Fleet Advisor collecte des données provenant de plusieurs environnements de base de données afin de fournir un aperçu de votre infrastructure de données. DMS Fleet Advisor collecte les données de vos serveurs d'analytique et de base de données sur site à partir d'un ou de plusieurs sites centraux, sans qu'il soit nécessaire de l'installer sur chaque ordinateur.

Actuellement, DMS Fleet Advisor prend en charge les serveurs de base de données Microsoft SQL Server, MySQL, Oracle et PostgreSQL.

Sur la base des données découvertes à partir de votre réseau, DMS Fleet Advisor établit un inventaire que vous pouvez passer en revue pour déterminer les serveurs de base de données et les objets à surveiller. Au fur et à mesure que des informations relatives à ces serveurs, bases de données et schémas sont collectées, vous pouvez analyser la faisabilité des migrations de bases de données que vous envisagez.

Migration de schéma et de code

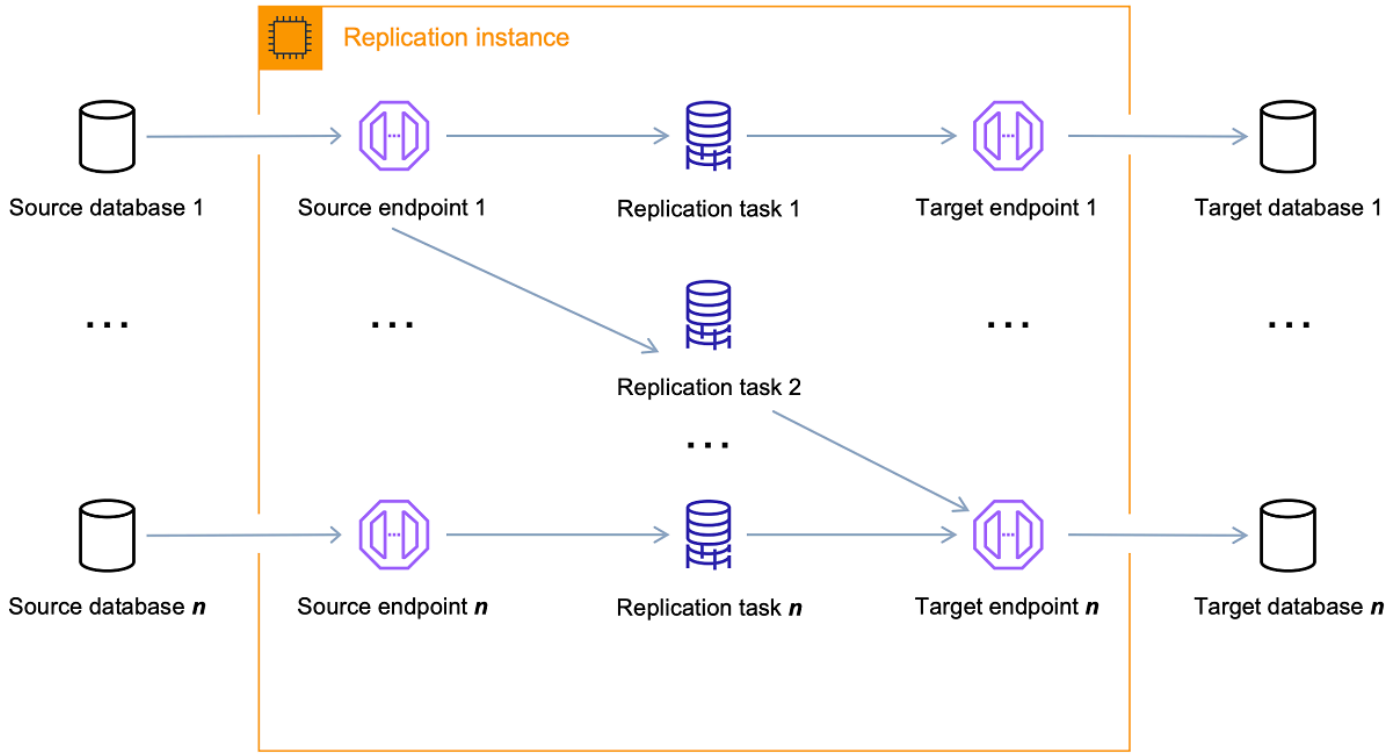
La conversion du schéma DMS dans AWS DMS rend les migrations de bases de données entre différents types de bases de données plus prévisibles. Vous pouvez utiliser la conversion de schéma DMS pour évaluer la complexité de votre migration pour votre fournisseur de données source, puis l'utiliser pour convertir les schémas de base de données et les objets de code. Vous pouvez alors appliquer le code converti à la base de données cible.

De manière générale, la conversion de schéma DMS fonctionne avec les trois composants suivants : les profils d'instance, les fournisseurs de données et les projets de migration. Un profil d'instance spécifie les paramètres réseau et de sécurité. Un fournisseur de données stocke les informations d'identification de connexion à la base de données. Un projet de migration contient des fournisseurs de données, un profil d'instance et des règles de migration. AWS DMS utilise des fournisseurs de données et un profil d'instance pour concevoir un processus de conversion des schémas de base de données et des objets de code.

Instance de réplication

À un niveau élevé, une instance de AWS DMS réplication est simplement une instance Amazon Elastic Compute Cloud (Amazon EC2) gérée qui héberge une ou plusieurs tâches de réplication.

La figure suivante montre un exemple d'instance de réplication qui exécute plusieurs tâches de réplication associées.



Une instance de réplication unique peut héberger une ou plusieurs tâches de réplication, en fonction des caractéristiques de votre migration et de la capacité du serveur de réplication. AWS DMS fournit une variété d'instances de réplication afin que vous puissiez choisir la configuration optimale pour votre cas d'utilisation. Pour plus d'informations sur les différentes classes d'instances de réplication, consultez Choisir l'instance de réplication AWS DMS adaptée à votre migration.

AWS DMS crée l'instance de réplication sur une instance Amazon EC2. Certaines des plus petites classes d'instance sont suffisantes pour tester le service ou effectuer de petites migrations. Si votre migration implique un grand nombre de tables, ou si vous prévoyez d'exécuter plusieurs tâches de réplication simultanées, vous devez envisager d'utiliser une des instances plus grandes. Cette approche est recommandée, car AWS DMS peut consommer une quantité importante de mémoire et d'UC.

En fonction de la classe d'instances Amazon EC2 que vous sélectionnez, votre instance de réplication dispose d'un stockage de données de 50 Go ou de 100 Go. Ce volume est généralement suffisant pour la plupart des clients. Toutefois, si votre migration implique de grandes transactions ou un important volume de changements de données, vous pouvez augmenter l'allocation de stockage de base. La capture des changements de données (CDC) peut entraîner l'écriture de données sur le disque selon la vitesse d'écriture des changements de la cible. Comme les fichiers journaux sont également écrits sur le disque, l'augmentation du

niveau de gravité de la journalisation entraîne également l'augmentation de la consommation de stockage.

AWS DMS peut fournir une haute disponibilité et un support de basculement à l'aide d'un déploiement multi-AZ. Dans un déploiement multi-AZ, AWS DMS provisionne et gère automatiquement une réplique de secours de l'instance de réplication dans une autre zone de disponibilité. L'instance de réplication principale est répliquée de manière synchrone vers le réplica de secours. Si l'instance de réplication principale échoue ou ne répond plus, l'instance de secours reprend toutes les tâches en cours avec une interruption minimale. Comme l'instance principale réplique constamment son état vers l'instance de secours, un déploiement multi-AZ occasionne une certaine dégradation des performances.

Pour des informations plus détaillées sur l'instance de AWS DMS réplication, consultezUtilisation d'une instance AWS DMS de réplication.

Plutôt que de créer et de gérer une instance de réplication, vous pouvez configurer votre AWS DMS réplication automatiquement à l'aide de AWS DMS Serverless. Pour plus d'informations, consultez Travailler avec AWS DMS Serverless.

Point de terminaison

AWS DMS utilise un point de terminaison pour accéder à votre banque de données source ou cible. Les informations de connexion spécifiques sont différentes en fonction de votre magasin de données mais, en général, vous fournissez les informations suivantes lorsque vous créez un point de terminaison :

- Type de point de terminaison : source ou cible.
- Type de moteur : type de moteur de base de données, tel qu'Oracle ou PostgreSQL.
- Nom du serveur : nom du serveur ou adresse IP AWS DMS accessible.
- Port : numéro de port utilisé pour les connexions de serveur de base de données.
- Chiffrement : mode SSL (Secure Socket Layer) si SSL est utilisé pour chiffrer la connexion.
- Informations d'identification : nom d'utilisateur et mot de passe pour un compte doté des droits d'accès requis.

Lorsque vous créez un point de terminaison à l'aide de la AWS DMS console, celle-ci vous demande de tester la connexion du point de terminaison. Le test doit réussir avant d'utiliser le point de terminaison dans une AWS DMS tâche. Comme les informations de connexion, les critères de test spécifiques sont différents pour différents types de moteur. En général, AWS DMS vérifie que la base de données existe pour le port et nom de serveur donnés, et que les

informations d'identification fournies permettent de se connecter à la base de données avec les privilèges nécessaires pour effectuer une migration. Si le test de connexion est réussi, AWS DMS télécharge et stocke les informations du schéma pour les utiliser ultérieurement lors de la configuration des tâches. Ces informations peuvent inclure des définitions de table, des définitions de clé primaire et des définitions de clé unique, par exemple.

Plusieurs tâches de réplication peuvent utiliser un seul point de terminaison. Par exemple, vous pouvez avoir deux applications logiquement distinctes, hébergées dans la même base de données source que vous souhaitez migrer séparément. Dans ce cas, vous créez deux tâches de réplication, une pour chaque ensemble de tables d'application. Vous pouvez utiliser le même AWS DMS point de terminaison pour les deux tâches.

Vous pouvez personnaliser le comportement d'un point de terminaison à l'aide des paramètres du point de terminaison. Les paramètres du point de terminaison peuvent contrôler divers comportements tels que le niveau de détail de la journalisation, la taille des fichiers et d'autres paramètres. Chaque type de moteur de stockage de données dispose de paramètres de point de terminaison différents. Vous pouvez trouver les paramètres de point de terminaison spécifiques pour chaque magasin de données dans la section source ou cible de ce magasin de données. Pour obtenir la liste des magasins de données source et cible pris en charge, consultez Sources pour AWS DMS et Objectifs pour AWS DMS.

Pour des informations plus détaillées sur les AWS DMS points de terminaison, consultezUtilisation des points de terminaison AWS DMS.

Tâches de réplication

Vous utilisez une tâche de AWS DMS réplication pour déplacer un ensemble de données du point de terminaison source vers le point de terminaison cible. La création d'une tâche de réplication est la dernière étape que vous devez effectuer avant de démarrer une migration.

Lorsque vous créez une tâche de réplication, vous spécifiez les paramètres de tâche suivants :

- Instance de réplication : l'instance utilisée pour héberger et exécuter la tâche
- Point de terminaison source
- Point de terminaison cible
- Options de type de migration, comme indiqué ci-après. Pour obtenir une explication complète des options de type de migration, reportez-vous à Création d'une tâche.
  - Chargement complet (Transférer les données existantes) : si vous pouvez vous permettre une interruption suffisamment longue pour copier vos données existantes, il est judicieux de

choisir cette option. Cette option migre simplement les données de votre base de données source vers votre base de données cible, en créant des tables le cas échéant.

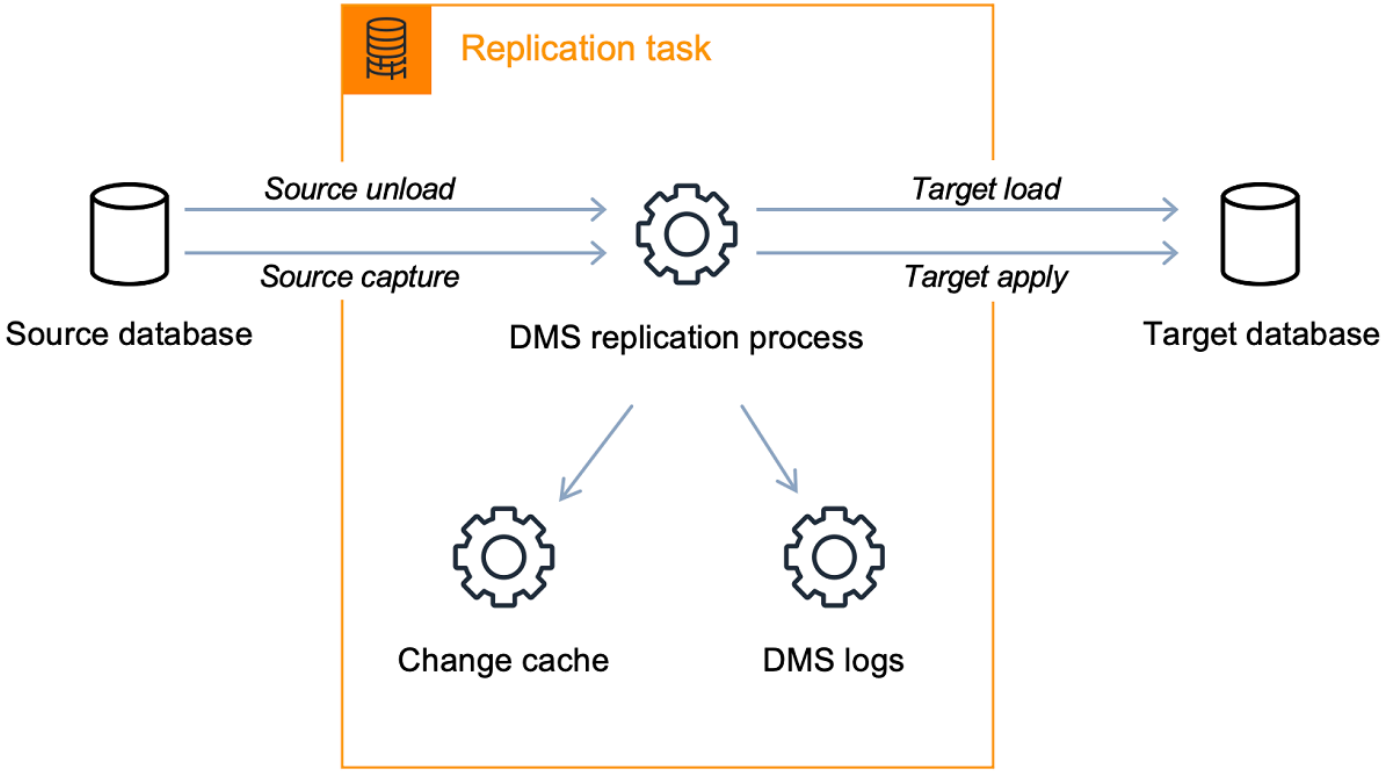
- Chargement complet + CDC (Migrer les données existantes et répliquer les modifications en cours) : cette option effectue un chargement complet des données tout en capturant les modifications apportées à la source. Une fois le chargement complet terminé, les modifications capturées sont appliquées à la cible. L'application des modifications finit par atteindre un état stable. Vous pouvez à ce stade fermer vos applications, laisser les modifications restantes être appliquées à la cible, puis redémarrer vos applications pointant vers la cible.
- CDC uniquement (Répliquer les modifications de données uniquement) : dans certains cas, il peut s'avérer plus efficace de copier les données existantes à l'aide d'une méthode autre que AWS DMS. Par exemple, dans une migration homogène, l'utilisation d'outils natifs d'importation/exportation peut se révéler plus efficace pour charger les données en masse. Dans ce cas, vous pouvez l'utiliser AWS DMS pour répliquer les modifications dès le début de votre chargement en masse afin de synchroniser vos bases de données source et cible.

- Options de mode de préparation de table cible, comme indiqué ci-après. Pour obtenir une présentation complète des modes de table cible, consultez Création d'une tâche.
  - Ne rien faire : AWS DMS suppose que les tables cibles sont précréées sur la cible.
  - Déposer les tables sur la cible : AWS DMS supprime et recrée les tables cibles.
  - Tronquer : si vous avez créé des tables sur la cible, AWS DMS les tronque avant le démarrage de la migration. S'il n'existe aucune table et que vous sélectionnez cette option, AWS DMS crée les tables manquantes.
- Options de mode LOB, comme indiqué ci-après. Pour obtenir une présentation complète des modes LOB, consultez Configuration du support LOB pour les bases de données sources dans une tâche AWS DMS.
  - Ne pas inclure les colonnes LOB : les colonnes LOB sont exclues de la migration.
  - Mode LOB complet : migrez des LOB complets, quelle que soit leur taille. AWS DMS migre les LOB par morceaux en morceaux contrôlés par le paramètre Max LOB Size. Ce mode est plus lent que le mode LOB limité.
  - Mode LOB limité : les LOB sont tronqués en fonction de la valeur spécifiée par le paramètre Taille maximale de LOB. Ce mode est plus rapide que le mode LOB complet.
- Mappages de tables : indique les tables à migrer et la façon dont elles sont migrées. Pour plus d'informations, consultez Utilisation du mappage de table pour spécifier des paramètres de tâche.

- Transformations de données, comme indiqué ci-après. Pour plus d'informations sur les transformations de données, consultez Spécification des règles de sélection de table et de transformation à l'aide de JSON.
  - Modification des noms de schéma, de table et de colonne.
  - Modification des noms d'espace de table (pour les points de terminaison cible Oracle).
  - Définition de clés principales et d'index uniques sur la cible.
- Validation des données
- CloudWatch Journalisation Amazon

Vous utilisez cette tâche pour migrer les données du point de terminaison source vers le point de terminaison cible ; le traitement des tâches s'effectue sur l'instance de réplication. Vous spécifiez les tables et les schémas à migrer, ainsi que tout traitement particulier, tel que les exigences de journalisation, les données des tables de contrôle et la gestion des erreurs.

Conceptuellement, une tâche de AWS DMS réplication exécute deux fonctions distinctes, comme le montre le schéma ci-dessous.



Le processus de chargement complet est simple à comprendre. Les données sont extraites de la source dans un mode d'extraction en bloc, puis chargées directement dans la cible. Vous pouvez

spécifier le nombre de tables à extraire et à charger en parallèle sur la AWS DMS console dans les paramètres avancés.

Pour plus d'informations sur AWS DMS les tâches, consultezUtilisation de tâches AWS DMS.

Réplication en cours ou capture des données modifiées

Vous pouvez également utiliser une AWS DMS tâche pour capturer les modifications continues apportées au magasin de données source pendant que vous migrez vos données vers une cible. Le processus de capture des modifications AWS DMS utilisé lors de la réplication des modifications en cours depuis un point de terminaison source collecte les modifications apportées aux journaux de base de données à l'aide de l'API native du moteur de base de données.

Dans le processus CDC, la tâche de réplication est conçue pour diffuser en continu les modifications de la source vers la cible, à l'aide de tampons en mémoire pour stocker les données en transit. Si les tampons en mémoire s'épuisent pour une raison quelconque, la tâche de réplication déverse les changements en attente dans le cache des changements sur le disque. Cela peut se produire, par exemple, si les modifications AWS DMS sont capturées à partir de la source plus rapidement qu'elles ne peuvent être appliquées à la cible. Dans ce cas, vous voyez la latence cible de la tâche dépasser la latence source de la tâche.

Vous pouvez vérifier cela en accédant à votre tâche sur la AWS DMS console et en ouvrant l'onglet Surveillance des tâches. Les LatencySource graphiques du CDC LatencyTarget et du CDC sont présentés au bas de la page. Si vous avez une tâche qui affiche la latence cible, il est probablement nécessaire d'effectuer quelques réglages sur le point de terminaison cible pour augmenter la vitesse d'application.

La tâche de réplication utilise également un stockage pour les journaux de tâches, comme décrit ci-dessus. L'espace disque qui est fourni préconfiguré avec votre instance de réplication est généralement suffisant pour la journalisation et les modifications déversées. Si vous avez besoin de plus d'espace disque, par exemple, lorsque vous utilisez le débogage détaillé pour étudier un problème de migration, vous pouvez modifier l'instance de réplication pour allouer plus d'espace.

Sources pour AWS DMS

Vous pouvez utiliser différents magasins de données sources dans différentes AWS DMS fonctionnalités. Les sections suivantes contiennent les listes des magasins de données sources pris en charge pour chaque AWS DMS fonctionnalité.

Rubriques

- Points de terminaison sources pour la migration des données
- Bases de données sources pour DMS Fleet Advisor
- Fournisseurs de données sources pour la conversion de schéma DMS
- Fournisseurs de données sources pour les migrations de données homogènes DMS

Points de terminaison sources pour la migration des données

Vous pouvez utiliser les magasins de données suivants comme points de terminaison source pour la migration des données à l'aide d' AWS DMS.

Bases de données sur site et d'instance EC2

- Oracle versions 10.2 et ultérieures (pour les versions 10.x), 11g et jusqu'à 12.2, 18c et 19c pour les éditions Enterprise, Standard, Standard One et Standard Two
- Microsoft SQL Server versions 2005, 2008, 2008R2, 2012, 2014, 2016, 2017, 2019 et 2022.
  - Les éditions Enterprise, Standard, Workgroup, Developer et Web prennent en charge la réplication à charge complète.
  - Les éditions Enterprise, Standard (version 2016 et ultérieure) et Developer prennent en charge la réplication CDC (continue) en plus du chargement complet.
  - L'édition Express n'est pas prise en charge.
- MySQL versions 5.5, 5.6, 5.7 et 8.0

  > **Note**
  >
  > Support de MySQL 8.0 en tant que source est disponible dans AWS DMS les versions 3.4.0 et supérieures, sauf lorsque la charge utile des transactions est compressée. Support pour Google Cloud for MySQL 8.0 en tant que source est disponible dans AWS DMS les versions 3.4.6 et supérieures.

- MariaDB (pris en charge en tant que source de données compatible MySQL) versions 10.0 (seulement les versions 10.0.24 et ultérieures), 10.2, 10.3, 10.4, 10.5 et 10.6.

  > **Note**
  >
  > Support pour MariaDB en tant que source est disponible dans AWS DMS toutes les versions où MySQL est supporté.

- PostgreSQL version 9.4 et ultérieures (pour les versions 9.x), 10.x, 11.x, 12.x, 13.x, 14.x et 15.x.

  > **Note**
  >
  > AWS DMS ne prend en charge PostgreSQL version 15.x que dans les versions 3.5.1 et supérieures.

- MongoDB versions 3.x, 4.0, 4.2, 4.4, 5.0 et 6.0.

  > **Note**
  >
  > AWS DMS les versions 3.5.0 et supérieures ne supportent pas les versions de MongoDB antérieures à 3.6.

- SAP Adaptive Server Enterprise (ASE) versions 12.5, 15, 15.5, 15.7, 16 et ultérieures
- Versions IBM Db2 pour Linux, UNIX et Windows (Db2 LUW) :
  - Version 9.7, tous les packs de correctifs
  - Version 10.1, tous les packs de correctifs
  - Version 10.5, tous les packs de correctifs, à l'exception du pack de correctifs 5
  - Version 11.1, tous les packs de correctifs
  - Version 11.5, Mods (0-8) avec uniquement le pack de correctifs 0
- IBM Db2 pour z/OS version 12

Services de base de données gérés par des tiers :

- Base de données Microsoft Azure SQL
- Microsoft Azure PostgreSQL Flexible Server versions 11.2, 12.15, 13.11, 14.8 et 15.3.
- Microsoft Azure MySQL Flexible Server versions 5.7 et 8.
- Google Cloud pour MySQL versions 5.6, 5.7 et 8.0.
- Google Cloud pour PostgreSQL versions 9.6, 10, 11, 12, 13, 14 et 15.
- OCI MySQL Heatwave version 8.0.34.

Bases de données d'instance Amazon RDS et Amazon Simple Storage Service (Amazon S3)

- Oracle versions 11g (versions 11.2.0.4 et ultérieures) et jusqu'à 12.2, 18c et 19c pour les éditions Enterprise, Standard, Standard One et Standard Two
- Microsoft SQL Server versions 2012, 2014, 2016, 2017, 2019 et 2022 pour les éditions Enterprise, Standard, Workgroup et Developer

> **Note**
>
> AWS DMS ne prend pas en charge SQL Server Express. L'édition Web est prise en charge pour la réplication à chargement complet uniquement.

- MySQL versions 5.5, 5.6, 5.7 et 8.0

> **Note**
>
> Support de MySQL 8.0 en tant que source est disponible dans AWS DMS les versions 3.4.0 et supérieures, sauf lorsque la charge utile des transactions est compressée.

- MariaDB (pris en charge en tant que source de données compatible MySQL) versions 10.0.24 à 10.0.28, 10.2, 10.3, 10.4, 10.5 et 10.6.

> **Note**
>
> Support pour MariaDB en tant que source est disponible dans AWS DMS toutes les versions où MySQL est supporté.

- PostgreSQL versions 10.x, 11.x, 12.x, 13.x, 14.x et 15.x.

> **Note**
>
> AWS DMS ne prend en charge PostgreSQL 15.x que dans les versions 3.5.1 et supérieures.

- Amazon Aurora avec compatibilité MySQL (prise en charge en tant que source de données compatible MySQL)
- Amazon Aurora avec compatibilité PostgreSQL (prise en charge en tant que source de données compatible PostgreSQL)

- Amazon S3
- Amazon DocumentDB (compatible avec MongoDB) versions 3.6, 4.0 et 5.0.
- Amazon RDS pour IBM Db2 LUW.

Pour plus d'informations sur l'utilisation d'une source spécifique, consultez la section Utilisation des AWS DMS points de terminaison.

Pour en savoir plus sur les points de terminaison cibles pris en charge, consultez Points de terminaison cibles pour la migration des données.

Bases de données sources pour DMS Fleet Advisor

DMS Fleet Advisor prend en charge les bases de données sources suivantes.

- Microsoft SQL Server versions 2012 et jusqu'à 2019
- MySQL versions 5.6 et jusqu'à 8
- Oracle versions 11g Release 2 et jusqu'à 12c, 19c et 21c
- PostgreSQL versions 9.6 et jusqu'à 13

Pour en savoir plus sur l'utilisation d'une source spécifique, consultez Création d'utilisateurs de base de données pour AWS DMS Fleet Advisor.

Pour obtenir la liste des bases de données utilisées par DMS Fleet Advisor pour générer des recommandations cibles, consultez Cibles pour DMS Fleet Advisor.

Fournisseurs de données sources pour la conversion de schéma DMS

La conversion de schéma DMS prend en charge les fournisseurs de données suivants en tant que sources pour vos projets de migration.

- Microsoft SQL Server versions 2008 R2, 2012, 2014, 2016, 2017 et 2019
- Oracle versions 10.2 et ultérieures, 11g et jusqu'à 12.2, 18c et 19c, et Oracle Data Warehouse
- PostgreSQL version 9.2 et supérieure
- MySQL version 5.5 et supérieure

Votre fournisseur de données sources peut être un moteur autogéré s'exécutant sur site ou sur une instance Amazon Elastic Compute Cloud (Amazon EC2).

Pour en savoir plus sur l'utilisation d'une source spécifique, consultez Création de fournisseurs de données sources dans la conversion de schéma DMS.

Pour en savoir plus sur les bases de données cibles prises en charge, consultez Fournisseurs de données cibles pour la conversion de schéma DMS.

Le AWS Schema Conversion Tool (AWS SCT) prend en charge un plus grand nombre de bases de données source et cible que la conversion de schéma DMS. Pour plus d'informations sur AWS SCT les bases de données compatibles, voir Qu'est-ce que le AWS Schema Conversion Tool.

Fournisseurs de données sources pour les migrations de données homogènes DMS

Vous pouvez utiliser les fournisseurs de données suivants en tant que sources pour des migrations de données homogènes.

- MySQL versions 5.7 et ultérieures
- MariaDB versions 10.2 et ultérieures
- PostgreSQL versions 10.4 à 14.x.
- MongoDB versions 4.x, 5.x, 6.0
- Amazon DocumentDB versions 3.6, 4.0, 5.0

Votre fournisseur de données source peut être un moteur autogéré s'exécutant sur site ou sur une instance Amazon EC2. Vous pouvez également utiliser une instance de base de données Amazon RDS en tant que fournisseur de données source.

Pour en savoir plus sur l'utilisation d'une source spécifique, consultez Création de fournisseurs de données sources pour des migrations de données homogènes dans AWS DMS.

Pour en savoir plus sur les bases de données cibles prises en charge, consultez Fournisseurs de données cibles pour les migrations de données homogènes DMS.

Objectifs pour AWS DMS

Vous pouvez utiliser différents magasins de données cibles dans différentes AWS DMS fonctionnalités. Les sections suivantes contiennent les listes des magasins de données cibles pris en charge pour chaque AWS DMS fonctionnalité.

Rubriques

- Points de terminaison cibles pour la migration des données
- Bases de données cibles pour DMS Fleet Advisor
- Fournisseurs de données cibles pour la conversion de schéma DMS
- Fournisseurs de données cibles pour les migrations de données homogènes DMS

Points de terminaison cibles pour la migration des données

Vous pouvez utiliser les magasins de données suivants comme points de terminaison cible pour la migration des données à l'aide d' AWS DMS.

Bases de données des instances sur site et Amazon EC2

- Oracle versions 10g, 11g, 12c, 18c et 19c pour les éditions Enterprise, Standard, Standard One et Standard Two.
- Microsoft SQL Server version 2005, 2008, 2008R2, 2012, 2014, 2016, 2017, 2019, et 2022 pour les éditions Enterprise, Standard, Workgroup et Developer

> **Note**
>
> AWS DMS ne prend pas en charge les éditions Web et Express de SQL Server.

- MySQL versions 5.5, 5.6, 5.7 et 8.0
- MariaDB (pris en charge en tant que cible de données compatible MySQL) versions 10.0.24 à 10.0.28, 10.2, 10.3, 10.4, 10.5 et 10.6.

> **Note**
>
> Support pour MariaDB en tant que cible est disponible dans AWS DMS toutes les versions où MySQL est supporté.

- PostgreSQL versions 9.4 et ultérieures (pour les versions 9.x), 10.x, 11.x, 12.x, 13.x, 14.x et 15.x.

> **Note**
>
> AWS DMS ne prend en charge PostgreSQL 15.x que dans les versions 3.5.1 et supérieures.

- SAP Adaptive Server Enterprise (ASE) versions 15, 15.5, 15.7, 16 et ultérieures
- Redis versions 6.x

Bases de données d'instances Amazon RDS, Amazon Redshift, Amazon Redshift Serverless, Amazon DynamoDB, Amazon S3, Amazon Service, Amazon pour Redis, Amazon OpenSearch Kinesis Data Streams, ElastiCache Amazon DocumentDB, Amazon Neptune et Apache Kafka

- Oracle version 11g (versions 11.2.0.3.v1 et ultérieures), 12c, 18c et 19c pour les éditions Enterprise, Standard, Standard One et Standard Two.
- Microsoft SQL Server versions 2012, 2014, 2016, 2017, 2019 et 2022 pour les éditions Enterprise, Standard, Workgroup et Developer

> **Note**
>
> AWS DMS ne prend pas en charge les éditions Web et Express de SQL Server.

- MySQL versions 5.5, 5.6, 5.7 et 8.0
- MariaDB (pris en charge en tant que cible de données compatible MySQL) versions 10.0.24 à 10.0.28, 10.2, 10.3, 10.4, 10.5 et 10.6.

> **Note**
>
> Support pour MariaDB en tant que cible est disponible dans AWS DMS toutes les versions où MySQL est supporté.

- PostgreSQL versions 10.x, 11.x, 12.x, 13.x, 14.x et 15.x

> **Note**
>
> AWS DMS ne prend en charge PostgreSQL 15.x que dans les versions 3.5.1 et supérieures.

- IBM Db2 LUW versions 11.1 et 11.5
- Amazon Aurora MySQL-Compatible Edition
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon Aurora sans serveur v2
- Amazon Redshift
- Amazon Redshift sans serveur
- Amazon S3
- Amazon DynamoDB
- Amazon OpenSearch Service
- Amazon ElastiCache pour Redis
- Amazon Kinesis Data Streams
- Amazon DocumentDB (compatible avec MongoDB)
- Amazon Neptune
- Apache Kafka : Amazon Managed Streaming for Apache Kafka (Amazon MSK) et Apache Kafka autogéré
- Babelfish (versions 3.2.0 et ultérieures) pour Aurora PostgreSQL (versions 15.3/14.8 et ultérieures)

Pour plus d'informations sur l'utilisation d'une cible spécifique, consultez la section Utilisation des AWS DMS points de terminaison.

Pour en savoir plus sur les points de terminaison sources pris en charge, consultez Points de terminaison sources pour la migration des données.

Bases de données cibles pour DMS Fleet Advisor

DMS Fleet Advisor génère des recommandations cibles à l'aide de la dernière version des bases de données cibles suivantes.

- Amazon Aurora MySQL
- Amazon Aurora PostgreSQL
- Amazon RDS for MySQL
- Amazon RDS for Oracle

- Amazon RDS for PostgreSQL
- Amazon RDS for SQL Server

Pour en savoir plus sur les recommandations cibles dans DMS Fleet Advisor, consultez Utilisation de la fonctionnalité de recommandations cibles d'AWS DMS Fleet Advisor.

Pour en savoir plus sur les bases de données sources prises en charge, consultez Bases de données sources pour DMS Fleet Advisor.

Fournisseurs de données cibles pour la conversion de schéma DMS

La conversion de schéma DMS prend en charge les fournisseurs de données suivants en tant que cibles pour vos projets de migration.

- Amazon Aurora MySQL 8.0.23
- Amazon Aurora PostgreSQL 14.5
- Amazon RDS for MySQL 8.0.23
- Amazon RDS for PostgreSQL 14.x
- Amazon Redshift

Pour en savoir plus sur l'utilisation d'une cible spécifique, consultez Création de fournisseurs de données cibles dans la conversion de schéma DMS.

Pour en savoir plus sur les bases de données sources prises en charge, consultez Fournisseurs de données sources pour la conversion de schéma DMS.

Fournisseurs de données cibles pour les migrations de données homogènes DMS

Vous pouvez utiliser les fournisseurs de données suivants en tant que cibles pour des migrations de données homogènes.

- Amazon Aurora MySQL version 5.7 et ultérieures
- Amazon Aurora PostgreSQL versions 10.4 à 14x
- Amazon Aurora sans serveur v2

- Amazon RDS for MySQL versions 5.7 ultérieures
- Amazon RDS for MariaDB versions 10.2 et ultérieures
- Amazon RDS for PostgreSQL versions 10.4 à 14x
- Amazon DocumentDB versions 4.0, 5.0 et cluster Elastic DocumentDB

Pour en savoir plus sur l'utilisation d'une cible spécifique, consultez Création de fournisseurs de données cibles pour les migrations de données homogènes dans AWS DMS.

Pour en savoir plus sur les bases de données sources prises en charge, consultez Fournisseurs de données sources pour les migrations de données homogènes DMS.

Création d'un nom de ressource Amazon (ARN) pour AWS DMS

Si vous utilisez l' AWS DMS API AWS CLI or pour automatiser la migration de votre base de données, vous utilisez Amazon Resource Name (ARNs). Chaque ressource créée dans Amazon Web Services est identifiée par un ARN, qui est un identificateur unique. Si vous utilisez l' AWS DMS API AWS CLI or pour configurer la migration de votre base de données, vous devez fournir l'ARN de la ressource avec laquelle vous souhaitez travailler.

L'ARN d'une AWS DMS ressource utilise la syntaxe suivante :

`arn:aws:dms:`*`region`*`:`*`account number`*`:`*`resourcetype`*`:`*`resourcename`*

Dans cette syntaxe, les éléments suivants s'appliquent :

- *`region`*est l'ID de l' Région AWS endroit où la AWS DMS ressource a été créée, par exemple`us-west-2`.

  Le tableau suivant indique les Région AWS noms et les valeurs que vous devez utiliser lors de la création d'un ARN.

Région	Nom
Région Asie-Pacifique (Tokyo)	ap-northeast-1
Région Asie-Pacifique (Séoul)	ap-northeast-2
Région Asie-Pacifique (Mumbai)	ap-south-1

Région	Nom
Région Asie-Pacifique (Singapour)	ap-southeast-1
Région Asie-Pacifique (Sydney)	ap-southeast-2
Région Canada (Centre)	ca-central-1
Région Chine (Pékin)	cn-north-1
Région Chine (Ningxia)	cn-northwest-1
Région Europe (Stockholm)	eu-north-1
Europe (Milan) Region	eu-south-1
Région UE (Francfort)	eu-central-1
Région Europe (Irlande)	eu-west-1
Région UE (Londres)	eu-west-2
Région Europe (Paris)	eu-west-3
Région Amérique du Sud (São Paulo)	sa-east-1
Région USA Est (Virginie du Nord)	us-east-1
Région US East (Ohio)	us-east-2
Région US West (N. California)	us-west-1
Région USA Ouest (Oregon)	us-west-2

- *`account number`* est votre numéro de compte sans les tirets. Pour trouver votre numéro de compte, connectez-vous à votre AWS compte sur http://aws.amazon.com, choisissez Mon compte/console, puis sélectionnez Mon compte.
- *`resourcetype`*est le type de AWS DMS ressource.

  Le tableau suivant indique les types de ressources à utiliser lors de la création d'un ARN pour une AWS DMS ressource particulière.

AWS DMS type de ressource	Format ARN
Instance de réplication	`arn:aws:dms:` *`region`*`:` *`account`*`:rep:` *`resourcename`*
Point de terminaison	`arn:aws:dms:` *`region`*`:`*`account`*`:endpoint:` *`resourcename`*
Tâche de réplication	`arn:aws:dms:` *`region`*`:`*`account`*`:task:`*`resourcename`*
Groupe de sous-réseaux	`arn:aws:dms:` *`region`*`:`*`account`*`:subgrp:`*`resourcename`*

- *`resourcename`*est le nom de ressource attribué à la AWS DMS ressource. Il s'agit d'une chaîne générée de façon arbitraire.

Le tableau suivant présente des exemples d'ARN pour les AWS DMS ressources. Nous supposons ici un numéro de compte AWS égal à 123456789012, créé dans la région USA Est (Virginie du Nord) et ayant un nom de ressource.

Type de ressource	Exemple d'ARN
Instance de réplication	`arn:aws:dms:us-east-1:123456789012:rep:QLXQZ64MH7CXF4QCQMGRVYVXAI`
Point de terminaison	`arn:aws:dms:us-east-1:123456789012:endpoint:D3HMZ2IGUCGFF3NTAXUXGF6S5A`
Tâche de réplication	`arn:aws:dms:us-east-1:123456789012:task:2PVREMWNPGYJCVU2IBPTOYTIV4`
Groupe de sous-réseaux	`arn:aws:dms:us-east-1:123456789012:subgrp:test-tag-grp`

Utilisation AWS DMS avec d'autres AWS services

Vous pouvez l'utiliser AWS DMS avec plusieurs autres AWS services :

- Vous pouvez utiliser une instance Amazon EC2 ou une instance de base de données Amazon RDS comme cible pour la migration des données.
- Vous pouvez utiliser le AWS Schema Conversion Tool (AWS SCT) pour convertir votre schéma source et votre code SQL en un schéma cible et un code SQL équivalents.
- Vous pouvez utiliser Amazon S3 comme site de stockage pour vos données ou l'utiliser comme étape intermédiaire lors de la migration de grandes quantités de données.
- Vous pouvez l'utiliser AWS CloudFormation pour configurer vos AWS ressources pour la gestion ou le déploiement de l'infrastructure. Par exemple, vous pouvez provisionner AWS DMS des ressources telles que des instances de réplication, des tâches, des certificats et des points de terminaison. Vous créez un modèle qui décrit toutes les AWS ressources que vous souhaitez, et qui AWS CloudFormation fournit et configure ces ressources pour vous.

AWS DMS support pour AWS CloudFormation

Vous pouvez provisionner AWS DMS des ressources à l'aide de AWS CloudFormation. AWS CloudFormation est un service qui vous aide à modéliser et à configurer vos AWS ressources pour la gestion ou le déploiement de l'infrastructure. Par exemple, vous pouvez provisionner AWS DMS des ressources telles que des instances de réplication, des tâches, des certificats et des points de terminaison. Vous créez un modèle qui décrit toutes les AWS ressources que vous souhaitez et qui AWS CloudFormation fournit et configure ces ressources pour vous.

En tant que développeur ou administrateur système, vous pouvez créer et gérer les collections de ces ressources, que vous pouvez, ensuite, utiliser pour les tâches de migration répétitives ou le déploiement des ressources dans votre organisation. Pour plus d'informations AWS CloudFormation, voir les AWS CloudFormation concepts dans le guide de AWS CloudFormation l'utilisateur.

AWS DMS prend en charge la création des AWS DMS ressources suivantes à l'aide de AWS CloudFormation :

- AWS::DMS::Certificate
- AWS::DMS::Endpoint
- AWS::DMS::EventSubscription
- AWS::DMS::ReplicationInstance
- AWS::DMS::ReplicationSubnetGroupe
- AWS::DMS::ReplicationTask

Démarrer avec AWS Database Migration Service

Dans le didacticiel suivant, vous découvrirez comment effectuer une migration de base de données à l'aide de AWS Database Migration Service (AWS DMS).

Pour effectuer une migration de base de données, procédez comme suit :

1. Configurez votre compte AWS en suivant les étapes détaillées dans Configuration pour AWS Database Migration Service.
2. Créez vos exemples de bases de données et un client Amazon EC2 pour remplir la base de données source et tester la réplication. De plus, créez un cloud privé virtuel (VPC) basé sur le service Amazon Virtual Private Cloud (Amazon VPC) qui contiendra les ressources du didacticiel. Pour créer ces ressources, suivez les étapes détaillées dans Prérequis pour AWS Database Migration Service.
3. Remplissez la base de données source à l'aide d'un exemple de script de création de base de données.
4. Utilisez la conversion de schéma DMS ou l'AWS Schema Conversion Tool (AWS SCT) pour convertir le schéma de la base de données source vers la base de données cible. Pour utiliser la conversion de schéma DMS, suivez les étapes détaillées dans Bien démarrer avec la conversion de schéma DMS. Pour convertir le schéma avec AWS SCT, suivez les étapes décrites dans Migration du schéma.
5. Créez une instance de réplication pour effectuer tous les processus de la migration. Pour ce faire et effectuer les tâches suivantes, suivez les étapes décrites dans Réplication.
6. Spécifiez les points de terminaison des bases de données source et cible. Pour de plus amples informations sur la création de points de terminaison, consultez Création de points de terminaison source et cible.
7. Créez une tâche pour définir les tables et les processus de réplication à utiliser, et démarrez la réplication. Pour en savoir plus sur la création de tâches de migration de base de données, consultez Création d'une tâche.
8. Vérifiez que la réplication fonctionne en exécutant des requêtes sur la base de données cible.

Configuration pour AWS Database Migration Service

S'inscrire à un Compte AWS

Si vous n'avez pas de compte Compte AWS, procédez comme suit pour en créer un.

Pour s'inscrire à un Compte AWS

1. Ouvrez https://portal.aws.amazon.com/billing/signup.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous souscrivez à un Compte AWS, un Utilisateur racine d'un compte AWS est créé. Par défaut, seul l'utilisateur root a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur administratif, et à uniquement utiliser l'utilisateur root pour effectuer tâches nécessitant un accès utilisateur root.

AWS vous envoie un e-mail de confirmation lorsque le processus d'inscription est terminé. Vous pouvez afficher l'activité en cours de votre compte et gérer votre compte à tout moment en accédant à https://aws.amazon.com/ et en cliquant sur Mon compte.

Création d'un utilisateur administratif

Une fois que vous êtes inscrit à un Compte AWS, sécurisez votre Utilisateur racine d'un compte AWS, activez AWS IAM Identity Center, puis créez un utilisateur administratif afin de ne pas employer l'utilisateur root pour les tâches quotidiennes.

Sécurisation de votre Utilisateur racine d'un compte AWS

1. Connectez-vous à la AWS Management Console en tant que propriétaire du compte en sélectionnant Root user (Utilisateur racine) et en saisissant l'adresse e-mail de Compte AWS. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur root, consultez Connexion en tant qu'utilisateur root dans le Guide de l'utilisateur Connexion à AWS.
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur root.

Pour obtenir des instructions, consultez Activation d'un dispositif MFA virtuel pour l'utilisateur root de votre Compte AWS (console) dans le Guide de l'utilisateur IAM.

Création d'un utilisateur administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d'AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center.

2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur administratif.

   Pour profiter d'un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, consultez Configuration de l'accès utilisateur avec le Répertoire IAM Identity Center par défaut dans le Guide de l'utilisateur AWS IAM Identity Center.

Connexion en tant qu'utilisateur administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

  Pour obtenir de l'aide pour vous connecter à l'aide d'un utilisateur IAM Identity Center, consultez Connexion au portail d'accès AWS dans le Guide de l'utilisateur Connexion à AWS.

Prérequis pour AWS Database Migration Service

Dans cette section, vous découvrirez les tâches prérequises pour AWS DMS, telles que la configuration des bases de données source et cible. Dans le cadre de ces tâches, vous pouvez également configurer un cloud privé virtuel (VPC) basé sur le service Amazon VPC, qui contiendra vos ressources. De plus, vous configurez une instance Amazon EC2 que vous utilisez pour remplir la base de données source et vérifier la réplication sur la base de données cible.

> **Note**
>
> Le remplissage de la base de données source prend jusqu'à 45 minutes.

Pour ce didacticiel, vous allez créer une base de données MariaDB en tant que source et une base de données PostgreSQL en tant que cible. Pour illustrer la réplication, ce scénario utilise des moteurs de base de données peu coûteux et couramment utilisés. L'utilisation de différents moteurs de base de données illustre les fonctionnalités AWS DMS de migration de données entre des plateformes hétérogènes.

Les ressources de ce didacticiel utilisent la région USA Ouest (Oregon). Si vous souhaitez utiliser une autre région AWS, spécifiez la région choisie à la place de la région USA Ouest (Oregon) partout où elle figure.

> **Note**
>
> Par souci de simplicité, les bases de données que vous créez pour ce didacticiel n'utilisent pas de chiffrement ni d'autres fonctionnalités de sécurité avancées. Vous devez utiliser des fonctionnalités de sécurité pour garantir la sécurité de vos bases de données de production. Pour plus d'informations, consultez Sécurité dans Amazon RDS.

Pour connaître les étapes prérequises, consultez les rubriques suivantes.

Rubriques

- Création d'un VPC
- Création de groupes de paramètres Amazon RDS
- Création de la base de données Amazon RDS source
- Création de la base de données Amazon RDS cible
- Création d'un client Amazon EC2
- Remplissage de la base de données source

Création d'un VPC

Dans cette section, vous allez créer un VPC qui contiendra vos ressources AWS. L'utilisation d'un VPC est une bonne pratique lorsque vous utilisez des ressources AWS, afin que vos bases de données, vos instances Amazon EC2, vos groupes de sécurité, etc., soient organisés de manière logique et sécurisés.

L'utilisation d'un VPC pour les ressources du didacticiel garantit également que vous supprimerez toutes les ressources que vous allez utiliser, une fois le didacticiel terminé. Vous devez supprimer toutes les ressources qu'un VPC contient avant de pouvoir supprimer le VPC.

Pour créer un VPC à utiliser avec AWS DMS

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. Dans le volet de navigation, choisissez Tableau de bord du VPC, puis Créer un VPC.
3. Sur la page Créer un VPC, définissez les options suivantes :

   - Ressources à créer : VPC et plus encore
   - Génération automatique d'identifications de noms : choisissez Génération automatique, puis entrez **`DMSVPC`**.
   - Bloc IPv4 : **`10.0.1.0/24`**
   - Bloc CIDR IPv6 : Pas de bloc CIDR IPv6
   - Location : Par défaut
   - Nombre de zones de disponibilité : 2
   - Nombre de sous-réseaux publics : 2
   - Nombre de sous-réseaux privés : 2
   - Passerelles NAT ($) : Aucune
   - Points de terminaison des VPC : Aucun

   Sélectionnez Create VPC (Créer un VPC).
4. Dans le volet de navigation, choisissez Vos VPC. Notez l'ID du VPC pour DMSVPC.
5. Dans le volet de navigation, choisissez Groupes de sécurité.
6. Choisissez le groupe nommé par défaut dont l'ID de VPC correspond à l'ID que vous avez noté pour DMSVPC.
7. Choisissez l'onglet Règles entrantes, puis Modifier les règles entrantes.
8. Choisissez Ajouter une règle. Ajoutez une règle de type MySQL/Aurora et choisissez Anywhere-IPv4 pour Source.
9. Choisissez à nouveau Ajouter une règle. Ajoutez une règle de type PostgreSQL et choisissez Anywhere-IPv4 pour Source.
10. Sélectionnez Enregistrer les règles.

Création de groupes de paramètres Amazon RDS

Pour spécifier les paramètres de vos bases de données source et cible pour AWS DMS, utilisez les groupes de paramètres Amazon RDS. Pour autoriser la réplication initiale et continue entre vos bases de données, assurez-vous de configurer les éléments suivants :

- Le journal binaire de la base de données source, afin qu'AWS DMS puisse déterminer les mises à jour incrémentielles qu'il doit répliquer.
- Le rôle de réplication de la base de données cible, afin qu'AWS DMS ignore les contraintes liées aux clés étrangères lors du transfert de données initial. Avec ce paramètre, AWS DMS peut migrer des données dans le désordre.

Pour créer des groupes de paramètres à utiliser avec AWS DMS

1. Ouvrez la console Amazon RDS à l'adresse https://console.aws.amazon.com/rds/.
2. Dans le volet de navigation, choisissez Groupes de paramètres.
3. Sur la page Groupes de paramètres, choisissez Créer un groupe de paramètres.
4. Sur la page Créer un groupe de paramètres, entrez les paramètres suivants :

   - Famille de groupes de paramètres : mariadb10.6
   - Group name (Nom de groupe: **`dms-mariadb-parameters`**
   - Description: **`Group for specifying binary log settings for replication`**

   Choisissez Créer.
5. Sur la page Groupes de paramètres, choisissez dms-mariadb-parameters, puis sur la dms-mariadb-parameterspage, sélectionnez Modifier.
6. Définissez les paramètres suivants sur les valeurs suivantes :

   - binlog_checksum : NONE
   - binlog_format : ROW

   Sélectionnez Enregistrer les modifications.
7. Sur la page Groupes de paramètres, choisissez à nouveau Créer un groupe de paramètres.
8. Sur la page Créer un groupe de paramètres, entrez les paramètres suivants :

- Famille de groupes de paramètres : postgres13
- Group name (Nom de groupe: **`dms-postgresql-parameters`**
- Description: **`Group for specifying role setting for replication`**

Choisissez Créer.

9. Sur la page Groupes de paramètres, sélectionnez dms-postgresql-parameters.
10. Sur la dms-postgresql-parameterspage, choisissez Modifier et définissez le paramètre session_replication_role sur replica. Notez que le paramètre session_replication_role ne figure pas sur la première page des paramètres. Utilisez les contrôles de pagination ou le champ de recherche pour rechercher le paramètre.
11. Sélectionnez Enregistrer les modifications.

Création de la base de données Amazon RDS source

Utilisez la procédure suivante pour créer la base de données Amazon RDS source.

Pour créer la base de données Amazon RDS for MariaDB source

1. Ouvrez la console Amazon RDS à l'adresse https://console.aws.amazon.com/rds/.
2. Sur la page Tableau de bord, choisissez Créer une base de données dans la section Base de données. Ne choisissez pas Créer une base de données dans la section Essayer la nouvelle option de déploiement Amazon RDS Multi-AZ pour MySQL et PostgreSQL en haut de la page.
3. Sur la page Créer une base de données, définissez les options suivantes :

   - Choisir une méthode de création de bases de données : choisissez Création standard.
   - Options de moteur : pour Type de moteur, choisissez MariaDB. Pour Version, laissez la version MariaDB 10.6.14 sélectionnée.
   - Modèles : choisissez Dev/Test.
   - Paramètres :
     - Identifiant d'instance de base de données : entrez **`dms-mariadb`**.
     - Dans la section Configuration des informations d'identification, entrez ce qui suit :
       - Identifiant principal : laissez **`admin`**.

    - Laissez l'option Gérer les informations d'identification principales dans AWS Secrets Manager non sélectionnée.
    - Génération automatique d'un mot de passe : laissez cette option désactivée.
    - Mot de passe principal : entrez **`changeit`**.
    - Confirmer le mot de passe : entrez **`changeit`** à nouveau.
- Configuration de l'instance :
    - Classe d'instance de base de données : laissez l'option Classes standard sélectionnée.
    - Pour Classe d'instance de base de données, choisissez db.m5.large.
- Stockage :
    - Décochez la case Activer la mise à l'échelle automatique du stockage.
    - Laissez le reste des paramètres tels qu'ils sont.
- Disponibilité et durabilité : laissez l'option Ne pas créer une instance de secours sélectionnée.
- Connectivité :
    - Ressource de calcul : laissez l'option Ne pas se connecter à une ressource de calcul EC2.
    - Type de réseau : laissez l'option IPv4 sélectionnée.
    - Cloud privé virtuel : DMSVPC-vpc
    - Accès public : Oui Vous devez activer l'accès public pour utiliser l'AWS Schema Conversion Tool.
    - Zone de disponibilité : us-west-2a
    - Laissez le reste des paramètres tels qu'ils sont.
- Authentification de base de données : laissez l'option Authentification par mot de passe sélectionnée.
- Sous Surveillance, décochez la case Activer Performance Insights. Développez la section Configuration supplémentaire et décochez la case Activer la surveillance améliorée.
- Développez Configuration supplémentaire :
    - Sous Options de base de données, entrez **`dms_sample`** pour Nom initial de base de données.
    - Sous Groupe de paramètres de base de données, sélectionnez dms-mariadb-parameters.
    - Pour Groupe d'options, laissez l'option default:mariadb-10-6 sélectionnée.
    - Sous Backup (Sauvegarde), procédez comme suit :

- Laissez l'option Activer les sauvegardes automatiques sélectionnée. Les sauvegardes automatiques de la base de données source doivent être activées pour prendre en charge la réplication continue.
- Pour Période de rétention des sauvegardes, choisissez 1 jour.
- Pour Fenêtre de sauvegarde, laissez l'option Aucune préférence sélectionnée.
- Décochez la case Copier les balises dans les instantanés.
- Laissez la case Activer la réplication dans une autre région AWS désactivée.

- Sous Chiffrement, décochez la case Activer le chiffrement.
- Laissez la section Exportations des journaux telle quelle.
- Sous Maintenance, décochez la case Activer la mise à niveau automatique des versions mineures et laissez le paramètre Fenêtre de maintenance défini sur Aucune préférence.
- Laissez la case Activer la protection contre la suppression désactivée.

4. Choisissez Create database (Créer une base de données).

Création de la base de données Amazon RDS cible

Répétez la procédure précédente pour créer la base de données Amazon RDS cible, avec les modifications suivantes.

Pour créer la base de données RDS for PostgreSQL cible

1. Répétez les étapes 1 et 2 de la procédure précédente.
2. Sur la page Créer une base de données, définissez les mêmes options, à l'exception des suivantes :

   a. Pour Options de moteur, choisissez PostgreSQL.

   b. Pour Version, choisissez PostgreSQL 13.7-R1

   c. Pour l'identifiant de l'instance DB, saisissez **`dms-postgresql`**.

   d. Pour Identifiant principal, laissez **`postgres`** sélectionné.

   e. Pour le groupe de paramètres de base de données, choisissez dms-postgresql-parameters.

   f. Désactivez l'option Activer les sauvegardes automatiques.

3. Choisissez Create database (Créer une base de données).

Création d'un client Amazon EC2

Dans cette section, vous allez créer un client Amazon EC2. Vous utilisez ce client pour remplir la base de données source avec des données à répliquer. Vous utilisez également ce client pour vérifier la réplication en exécutant des requêtes sur la base de données cible.

L'utilisation d'un client Amazon EC2 pour accéder à vos bases de données présente les avantages suivants par rapport à un accès via Internet à vos bases de données :

- Vous pouvez restreindre l'accès à vos bases de données aux clients qui se trouvent dans le même VPC.
- Nous avons confirmé que les outils que vous utilisez dans ce didacticiel fonctionnent et sont faciles à installer sur Amazon Linux 2023, ce que nous recommandons pour ce didacticiel.
- Les opérations de données entre les composants d'un VPC sont généralement plus performantes que celles effectuées via Internet.

Pour créer et configurer un client Amazon EC2 afin de remplir la base de données source

1. Ouvrez la console Amazon EC2 à l'adresse https://console.aws.amazon.com/ec2/.
2. Dans le tableau de bord, choisissez Lancer une instance.
3. Sur la page Lancer une instance, entrez les valeurs suivantes :

    a. Dans la section Noms et balises, entrez **`DMSClient`** pour Nom.

    b. Dans la section Images d'applications et de systèmes d'exploitation (Amazon Machine Image), laissez les paramètres tels qu'ils sont.

    c. Dans la section Type d'instance, choisissez t2.xlarge.

    d. Sous Paire de clés (connexion), choisissez Créer une nouvelle paire de clés.

    e. Sur la page Créer une paire de clés, entrez ce qui suit :

    - Nom de paire de clés : **`DMSKeyPair`**
    - Type de paire de clés : laissez RSA.
    - Format de fichier de clé privée : choisissez pem pour OpenSSH sur macOS ou Linux, ou ppk pour PuTTY sur Windows.

    Enregistrez le fichier de clé lorsque vous y êtes invité.

> **Note**
>
> Vous pouvez également utiliser une paire de clés Amazon EC2 existante au lieu d'en créer une nouvelle.

f. Dans la section Paramètres réseau, choisissez Modifier. Sélectionnez les paramètres suivants :

- VPC - obligatoire : choisissez le VPC avec l'ID que vous avez enregistré pour le VPC DMSVPC-vpc.
- Sous-réseau : choisissez le premier sous-réseau public.
- Attribuer automatiquement l'adresse IP publique : choisissez Activer.

Laissez le reste des paramètres tels qu'ils sont et choisissez Lancer l'instance.

Remplissage de la base de données source

Dans cette section, vous trouverez des points de terminaison pour vos bases de données source et cible à utiliser ultérieurement et vous utiliserez les outils suivants pour remplir la base de données source :

- Git, pour télécharger le script de remplissage de la base de données source.
- Client MariaDB, pour exécuter ce script.

Obtention des points de terminaison

Recherchez et notez les points de terminaison de vos instances de base de données RDS pour MariaDB et RDS pour PostgreSQL pour une utilisation ultérieure.

Pour rechercher les points de terminaison d'instance de base de données

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon RDS à l'adresse https://console.aws.amazon.com/rds/.
2. Dans le volet de navigation, choisissez Bases de données.
3. Choisissez la base de données dms-mariadb et notez la valeur de Point de terminaison pour la base de données.

4. Répétez les étapes précédentes pour la base de données dms-postgresql.

Remplissage de la base de données source

Connectez-vous ensuite à l'instance client, installez le logiciel nécessaire, téléchargez des exemples de scripts de base de données AWS depuis Git et exécutez ces scripts pour remplir la base de données source.

Pour remplir la base de données source

1. Connectez-vous à l'instance client à l'aide du nom d'hôte et de la clé publique que vous avez enregistrés lors des étapes précédentes.

   Pour plus d'informations sur la connexion à une instance Amazon EC2, consultez Accès aux instances dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

   > **Note**
   >
   > Si vous utilisez PuTTY, activez les keepalives TCP sur la page des paramètres de connexion afin que votre connexion ne soit pas interrompue suite à un délai d'inactivité.

2. Installez Git, MariaDB et PostgreSQL. Confirmez l'installation si nécessaire.

   ```
   $ sudo yum install git
   $ sudo dnf install mariadb105
   $ sudo dnf install postgresql15
   ```

3. Exécutez la commande suivante pour télécharger les scripts de création de base de données depuis GitHub.

   ```
   git clone https://github.com/aws-samples/aws-database-migration-samples.git
   ```

4. Passez au répertoire `aws-database-migration-samples/mysql/sampledb/v1/`.
5. Exécutez la commande suivante. Indiquez le point de terminaison pour votre instance RDS source que vous avez noté précédemment, par exemple `dms-mariadb.cdv5fbeyiy4e.us-east-1.rds.amazonaws.com`.

   ```
   mysql -h dms-mariadb.abcdefghij01.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
    dms_sample < ~/aws-database-migration-samples/mysql/sampledb/v1/install-rds.sql
   ```

6. Laissez le script de création de base de données s'exécuter. Le script prend jusqu'à 45 minutes pour créer le schéma et renseigner les données. Vous pouvez ignorer en toute sécurité les erreurs et les avertissements affichés par le script.

Migration du schéma source vers la base de données cible à l'aide d'AWS SCT

Dans cette section, vous allez utiliser l'AWS Schema Conversion Tool pour migrer votre schéma source vers la base de données cible. Vous pouvez également utiliser la conversion de schéma DMS pour convertir vos schémas de base de données source. Pour de plus amples informations, veuillez consulter Bien démarrer avec la conversion de schéma DMS.

Pour migrer le schéma source vers la base de données cible à l'aide d'AWS SCT

1. Installez la AWS Schema Conversion Tool. Pour plus d'informations, consultez Installation, vérification et mise à jour d'AWS SCT dans le Guide de l'utilisateur de l'Outil de conversion des schémas AWS.

   Lorsque vous téléchargez les pilotes JDBC pour MySQL et PostgreSQL, notez où vous les enregistrez, au cas où l'outil vous demanderait leurs emplacements.
2. Ouvrez la AWS Schema Conversion Tool. Choisissez Fichier, puis Nouveau projet.
3. Dans la fenêtre Nouveau projet, définissez les valeurs suivantes :

   - Définissez Nom du projet sur **`DMSProject`**.
   - Conservez Emplacement tel quel pour stocker votre projet AWS SCT dans le dossier par défaut.

   Sélectionnez OK.
4. Choisissez Ajouter une source pour ajouter une base de données MySQL source à votre projet et choisissez MySQL, puis Suivant.
5. Sur la page Ajouter une source, définissez les valeurs suivantes :

   - Nom de la connexion : **`source`**
   - Nom du serveur : entrez le point de terminaison de la base de données MySQL que vous avez noté précédemment.
   - Port du serveur : **`3306`**

- Nom d'utilisateur : **admin**
- Mot de passe : **changeit**

6. Choisissez Ajouter une cible pour ajouter une base de données Amazon RDS for PostgreSQL cible à votre projet, puis choisissez Amazon RDS for PostgreSQL. Choisissez Next (Suivant).
7. Sur la page Ajouter une cible, définissez les valeurs suivantes :

   - Nom de la connexion : **target**
   - Nom du serveur : entrez le point de terminaison de la base de données PostgreSQL que vous avez noté précédemment.
   - Port du serveur : **5432**
   - Base de données : entrez le nom de la base de données PostgreSQL.
   - Nom d'utilisateur : **postgres**
   - Mot de passe : **changeit**
8. Dans le volet de gauche, choisissez dms_sample sous Schémas. Dans le volet droit, choisissez la base de données Amazon RDS for PostgreSQL cible. Choisissez Créer un mappage.
   Vous pouvez ajouter plusieurs règles de mappage à un même projet AWS SCT. Pour plus d'informations sur les règles de mappage, consultez Création de règles de mappage.
9. Choisissez Vue principale.
10. Dans le volet de gauche, choisissez dms_sample sous Schémas. Ouvrez le menu contextuel (clic droit) et choisissez Convertir le schéma. Confirmez l'action.

    Une fois que l'outil a converti le schéma, le schéma dms_sample apparaît dans le volet de droite.
11. Dans le volet de droite, sous Schémas, ouvrez le menu contextuel (clic droit) de dms_sample et choisissez Appliquer à la base de données. Confirmez l'action.

Vérifiez que la migration du schéma est terminée. Procédez comme suit.

Pour vérifier la migration de votre schéma

1. Connectez-vous à votre client Amazon EC2.
2. Démarrez le client PSQL avec la commande suivante. Spécifiez le point de terminaison de la base de données PostgreSQL et fournissez le mot de passe de la base de données lorsque vous y êtes invité.

```
psql \
```

```
   --host=dms-postgresql.abcdefg12345.us-west-2.rds.amazonaws.com \
   --port=5432 \
   --username=postgres \
   --password \
   --dbname=dms_sample
```

3. Interrogez l'une des tables (vides) pour vérifier que l'AWS SCT a correctement appliqué le schéma.

```
dms_sample=> SELECT * from dms_sample.player;
 id | sport_team_id | last_name | first_name | full_name
----+---------------+-----------+------------+-----------
(0 rows)
```

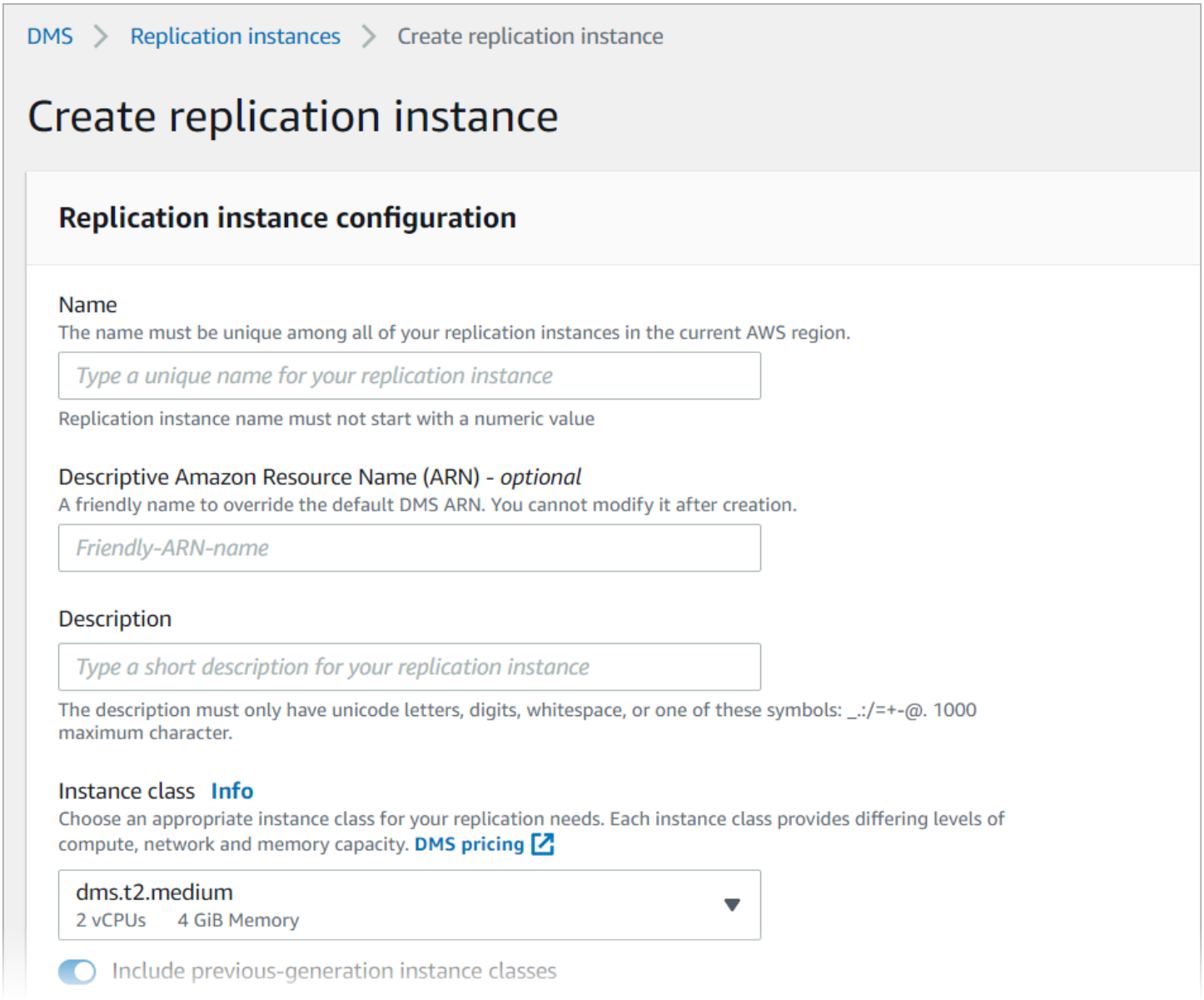
Configuration de la réplication pour AWS Database Migration Service

Dans cette rubrique, vous allez configurer la réplication entre les bases de données source et cible.

Étape 1 : Créer une instance de réplication à l'aide de la console AWS DMS

Pour commencer à utiliser AWS DMS, créez une instance de réplication.

Une instance de réplication effectue la migration effective des données entre les points de terminaison sources et cibles. L'instance nécessite suffisamment de stockage et de puissance de traitement pour effectuer les tâches qui migrent les données de la base de données source vers la base de données cible. La taille requise de cette instance de réplication dépend de la quantité de données à migrer et des tâches que l'instance doit effectuer. Pour plus d'informations sur les instances de réplication, consultez la page Utilisation d'une instance AWS DMS de réplication.



Pour créer une instance de réplication à l'aide de la console

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS DMS à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Dans le volet de navigation, choisissez Instances de réplication, puis Créer une instance de réplication.
3. Sur la page Créer une instance de réplication, spécifiez la configuration d'instance de réplication :

   a. Pour Name (Nom), saisissez **`DMS-instance`**.

b. Pour Description, entrez une brève description de l'instance de réplication (facultatif).

c. Pour Classe d'instance, laissez l'option dms.t3.medium sélectionnée.

   L'instance a besoin d'une capacité de stockage, d'une mise en réseau et d'une puissance de traitement suffisantes pour la migration. Pour plus d'informations sur la façon de choisir une classe d'instances, consultez Choisir l'instance de réplication AWS DMS adaptée à votre migration.

d. Pour Version du moteur, acceptez la valeur par défaut.

e. Pour Multi AZ, choisissez Charge de travail de développement ou de test (Mono-AZ).

f. Pour Stockage alloué (Gio), acceptez la valeur par défaut de 50 Gio.

   Dans AWS DMS, le stockage est principalement utilisé par les fichiers journaux et les transactions mises en cache. Pour les transactions mises en cache, le stockage est utilisé uniquement lorsque les transactions mises en cache doivent être écrites sur disque. Par conséquent, AWS DMS n'utilise pas une quantité importante de stockage.

g. Pour Type de réseau, choisissez IPv4.

h. Pour VPC, choisissez DMSVPC.

i. Pour Groupe de sous-réseaux de réplication, laissez le groupe de sous-réseaux de réplication actuellement sélectionné.

j. Désactivez Accessible publiquement.

4. Choisissez l'onglet Sécurité avancée et configuration réseau pour définir les valeurs des paramètres réseau et de chiffrement si vous en avez besoin :

   a. Pour Zone de disponibilité, choisissez us-west-2a.

   b. Pour Groupe(s) de sécurité VPC, choisissez le groupe de sécurité Par défaut s'il n'est pas déjà sélectionné.

   c. Pour AWS KMS key, laissez l'option (Par défaut) aws/dms sélectionnée.

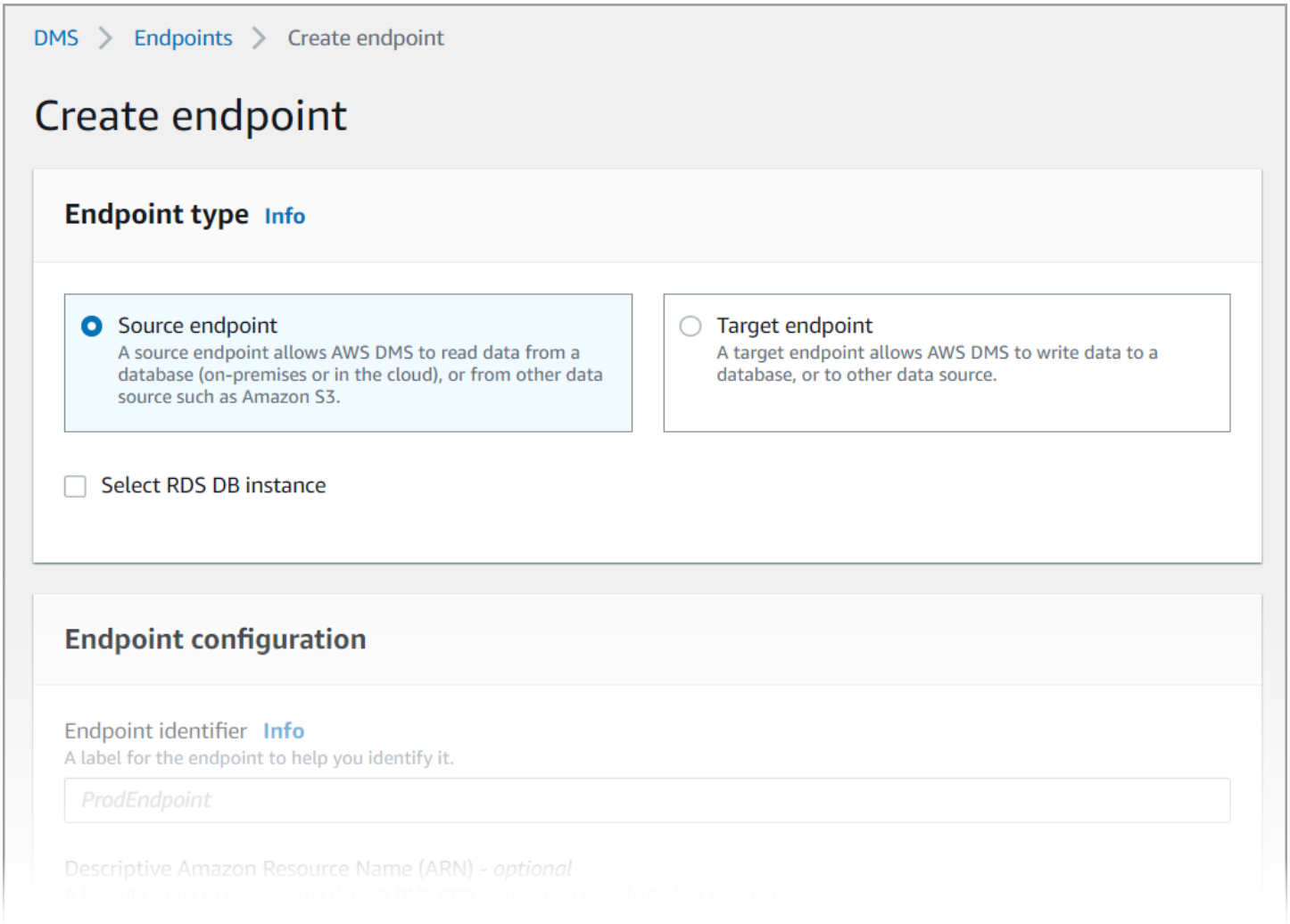
5. Laissez les paramètres de l'onglet Maintenance tels quels. Par défaut, une fenêtre de 30 minutes est sélectionnée de manière aléatoire dans un bloc de 8 heures pour chaque région AWS, se produisant un jour choisi au hasard dans la semaine.

6. Choisissez Créer.

AWS DMS crée une instance de réplication pour effectuer votre migration.

Étape 2 : Spécifier les points de terminaison sources et cibles

Pendant la création de votre instance de réplication, vous pouvez spécifier les points de terminaison des magasins de données sources et cibles pour les bases de données Amazon RDS que vous avez créées précédemment. Vous créez chaque point de terminaison séparément.



Pour spécifier un point de terminaison source et un point de terminaison de base de données à l'aide de la console AWS DMS

1. Dans la console, choisissez Points de terminaison dans le volet de navigation, puis choisissez Créer un point de terminaison.
2. Sur la page Créer un point de terminaison, choisissez le type de point de terminaison Source. Cochez la case Sélectionner une instance de base de données RDS et choisissez l'instance dms-mariadb.

3. Dans la section Configuration du point de terminaison, entrez **dms-mysql-source** pour Identifiant de point de terminaison.
4. Pour Moteur source, laissez MySQL sélectionné.
5. Pour Accès à la base de données du point de terminaison, choisissez Renseignez les informations d'accès manuellement. Vérifiez que les paramètres Port, Mode Secure Socket Layer (SSL), Nom d'utilisateur et Mot de passe sont corrects.
6. Choisissez l'onglet Tester connexion de point de terminaison (facultatif). Pour VPC, choisissez DMSVPC.
7. Pour Instance de réplication, laissez l'instance dms-instance sélectionnée.
8. Choisissez Exécuter un test.

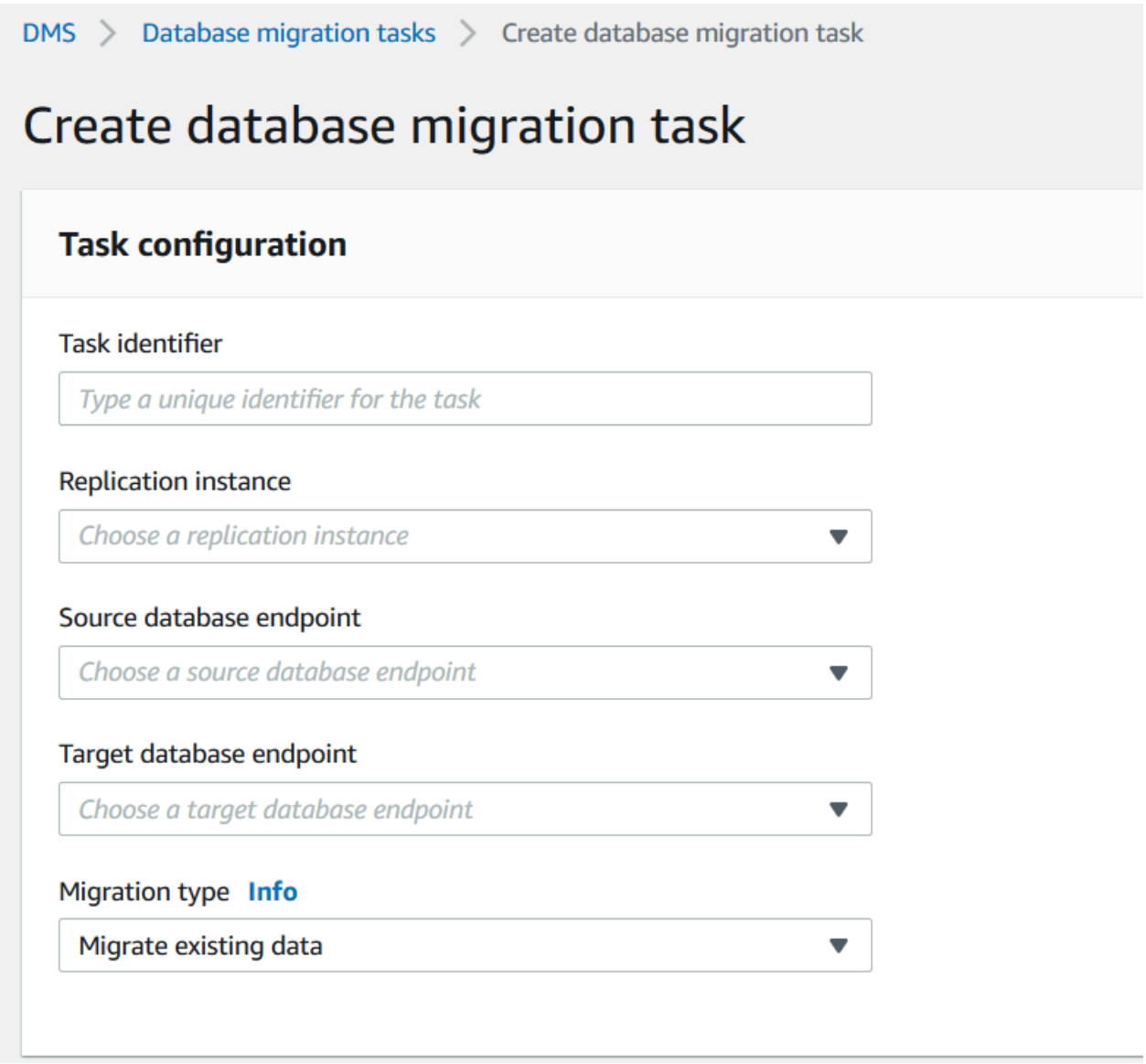
   Une fois que vous avez choisi Exécuter un test, AWS DMS crée le point de terminaison avec les détails que vous avez fournis et s'y connecte. Si la connexion échoue, modifiez la définition du point de terminaison et testez à nouveau la connexion. Vous pouvez également supprimer le point de terminaison manuellement.
9. Une fois le test réussi, choisissez Créer un point de terminaison.
10. Spécifiez un point de terminaison de base de données cible à l'aide de la console AWS DMS. Pour cela, répétez les étapes précédentes avec les paramètres suivants :

    - Type de point de terminaison : Point de terminaison cible
    - Instance RDS : dms-postgresql
    - Identifiant de point de terminaison : **dms-postgresql-target**
    - Moteur cible : laissez **PostgreSQL** sélectionné.

Lorsque vous avez terminé de fournir toutes les informations relatives à vos points de terminaison, AWS DMS crée vos points de terminaison sources et cibles à utiliser lors de la migration de la base de données.

Étape 3 : Créer une tâche et migrer les données

Dans cette étape, vous allez créer une tâche pour migrer les données entre les bases de données que vous avez créées.



Pour créer une tâche de migration et commencer votre migration de base de données

1. Dans le volet de navigation de la console, choisissez Tâches de migration de base de données, puis choisissez Créer une tâche. La page Créer une tâche de migration de base de données s'ouvre.
2. Dans la section Configuration de la tâche, spécifiez les options de tâche suivantes :
   - Identifiant de tâche : entrez **`dms-task`**.
   - Instance de réplication : choisissez votre instance de réplication (dms-instance-vpc-***<vpc id>***).
   - Point de terminaison de base de données source : choisissez dms-mysql-source.

- Point de terminaison de base de données cible : choisissez dms-postgresql-target.
- Type de migration : choisissez Migrer les données existantes et répliquer les modifications en cours.

3. Choisissez l'onglet Paramètres de tâche. Définissez les paramètres suivants :
   - Mode de préparation des tables cible : Ne rien faire
   - Arrêter la tâche après la fin du chargement complet : Ne pas arrêter
4. Choisissez l'onglet Mappages de table, puis développez Règles de sélection. Choisissez Ajouter une nouvelle règle de sélection. Définissez les paramètres suivants :
   - Schéma : Entrer un schéma
   - Nom du schéma : **`dms_sample`**
5. Choisissez l'onglet Configuration du démarrage de la tâche de migration, puis choisissez Automatiquement lors de la création.
6. Choisissez Créer tâche.

AWS DMS crée ensuite la tâche de migration et la démarre. La réplication de base de données initiale prend environ 10 minutes. Assurez-vous d'effectuer l'étape suivante du didacticiel avant qu'AWS DMS termine la migration des données.

Étape 4 : Tester la réplication

Dans cette section, vous insérez des données dans la base de données source pendant et après la réplication initiale, et vous interrogez la base de données cible pour récupérer les données insérées.

Pour tester la réplication

1. Assurez-vous que la tâche de migration de base de données affiche le statut En cours d'exécution, mais que la réplication de la base de données initiale, démarrée à l'étape précédente, n'est pas terminée.
2. Connectez-vous à votre client Amazon EC2 et démarrez le client MySQL à l'aide de la commande suivante. Fournissez le point de terminaison de la base de données MySQL.

```
mysql -h dms-mysql.abcdefg12345.us-west-2.rds.amazonaws.com -P 3306 -u admin -
pchangeit dms_sample
```

3. Exécutez la commande suivante pour insérer un enregistrement dans la base de données source.

```
MySQL [dms_sample]> insert person (full_name, last_name, first_name) VALUES ('Test
 User1', 'User1', 'Test');
Query OK, 1 row affected (0.00 sec)
```

4. Quittez le client MySQL.

```
MySQL [dms_sample]> exit
Bye
```

5. Avant la fin de la réplication, interrogez la base de données cible pour obtenir le nouvel enregistrement.

   Depuis l'instance Amazon EC2, connectez-vous à la base de données cible à l'aide de la commande suivante, en fournissant le point de terminaison de la base de données cible.

```
psql \
   --host=dms-postgresql.abcdefg12345.us-west-2.rds.amazonaws.com \
   --port=5432 \
   --username=postgres \
   --password \
   --dbname=dms_sample
```

   Fournissez le mot de passe (**changeit**) lorsque vous y êtes invité.

6. Avant la fin de la réplication, interrogez la base de données cible pour obtenir le nouvel enregistrement.

```
dms_sample=> select * from dms_sample.person where first_name = 'Test';
 id | full_name | last_name | first_name
----+-----------+-----------+------------
(0 rows)
```

7. Pendant l'exécution de la tâche de migration, vous pouvez surveiller la progression de la migration de la base de données :

   - Dans le volet de navigation de la console DMS, choisissez Tâches de migration de base de données.
   - Choisissez dms-task.

- Choisissez Statistiques de table.

Pour de plus amples informations sur la surveillance, veuillez consulter Surveillance des tâches AWS DMS.

8. Une fois la réplication terminée, interrogez à nouveau la base de données cible pour obtenir le nouvel enregistrement. AWS DMS migre le nouvel enregistrement une fois la réplication initiale terminée.

```
dms_sample=> select * from dms_sample.person where first_name = 'Test';
   id    | full_name  | last_name | first_name
---------+------------+-----------+------------
 7077784 | Test User1 | User1     | Test
(1 row)
```

9. Quittez le client psql.

```
dms_sample=> quit
```

10. Répétez l'étape 1 pour vous reconnecter à la base de données source.
11. Insérez un autre enregistrement dans la table `person`.

```
MySQL [dms_sample]> insert person (full_name, last_name, first_name) VALUES ('Test
 User2', 'User2', 'Test');
Query OK, 1 row affected (0.00 sec)
```

12. Répétez les étapes 3 et 4 pour vous déconnecter de la base de données source et vous connecter à la base de données cible.
13. Interrogez à nouveau la base de données cible pour les données répliquées.

```
dms_sample=> select * from dms_sample.person where first_name = 'Test';
   id    | full_name  | last_name | first_name
---------+------------+-----------+------------
 7077784 | Test User1 | User1     | Test
 7077785 | Test User2 | User2     | Test
(2 rows)
```

Étape 5 : Nettoyer les ressources AWS DMS

Une fois le didacticiel de mise en route terminé, vous pouvez supprimer les ressources que vous avez créées. Vous pouvez utiliser la console AWS pour les supprimer. Assurez-vous de supprimer les tâches de migration avant de supprimer l'instance de réplication et les points de terminaison.

Pour supprimer une tâche de migration à l'aide de la console

1. Dans le volet de navigation de la console AWS DMS, choisissez Tâches de migration de base de données.
2. Choisissez dms-task.
3. Sélectionnez Actions, Delete (Supprimer).

Pour supprimer une instance de réplication à l'aide de la console

1. Dans le volet de navigation de la console AWS DMS, choisissez Instances de réplication.
2. Choisissez DMS-instance.
3. Sélectionnez Actions, Delete (Supprimer).

AWS DMS supprime l'instance de réplication et la retire de la page Instances de réplication.

Pour supprimer les points de terminaison à l'aide de la console

1. Dans le volet de navigation de la console AWS DMS, choisissez Points de terminaison.
2. Choisissez dms-mysql-source.
3. Sélectionnez Actions, Delete (Supprimer).

Après avoir supprimé vos ressources AWS DMS, assurez-vous de supprimer également les ressources suivantes. Pour obtenir de l'aide concernant la suppression de ressources dans d'autres services, consultez la documentation de chaque service.

- Vos bases de données RDS.
- Vos groupes de paramètres de base de données RDS.
- Vos groupes de sous-réseaux RDS.
- Tous les journaux Amazon CloudWatch créés avec vos bases de données et l'instance de réplication.

- Groupes de sécurité créés pour votre réseau Amazon VPC et le client Amazon EC2. Assurez-vous de supprimer la règle entrante dans Par défaut pour les groupes de sécurité launch-wizard-1, ce qui est nécessaire pour que vous puissiez les supprimer.
- Votre client Amazon EC2.
- Votre réseau Amazon VPC.
- Votre paire de clés Amazon EC2 du client Amazon EC2.

Ressources supplémentaires pour l'utilisation d'AWS Database Migration Service

Plus loin dans ce guide, vous pourrez apprendre à utiliser AWS DMS pour migrer vos données vers et depuis la plupart des bases de données commerciales et open source les plus répandues.

Nous vous recommandons également de consulter les ressources suivantes lors de la préparation et de la réalisation d'un projet de migration de base de données :

- Guide de migration pas à pas AWS DMS : ce guide fournit les procédures pas à pas qui composent le processus de migration des données vers AWS.
- Référence d'API AWS DMS : cette référence décrit en détail toutes les opérations d'API pour AWS Database Migration Service.
- AWS CLI pour AWS DMS : cette référence fournit des informations sur l'utilisation d'AWS Command Line Interface (AWS CLI) avec AWS DMS.

Découverte et évaluation des bases de données pour la migration avec AWS DMS Fleet Advisor

Vous pouvez utiliser DMS Fleet Advisor pour collecter des métadonnées et des métriques de performance à partir de plusieurs environnements de base de données. Ces métriques collectées fournissent des informations sur votre infrastructure de données. DMS Fleet Advisor collecte des métadonnées et des métriques à partir de votre base de données sur site et de vos serveurs d'analytique, à partir d'un ou de plusieurs emplacements centraux, sans qu'il soit nécessaire de l'installer sur chaque ordinateur. Actuellement, DMS Fleet Advisor prend en charge la découverte et la collecte de métriques pour les serveurs de base de données Microsoft SQL Server, MySQL, Oracle et PostgreSQL.

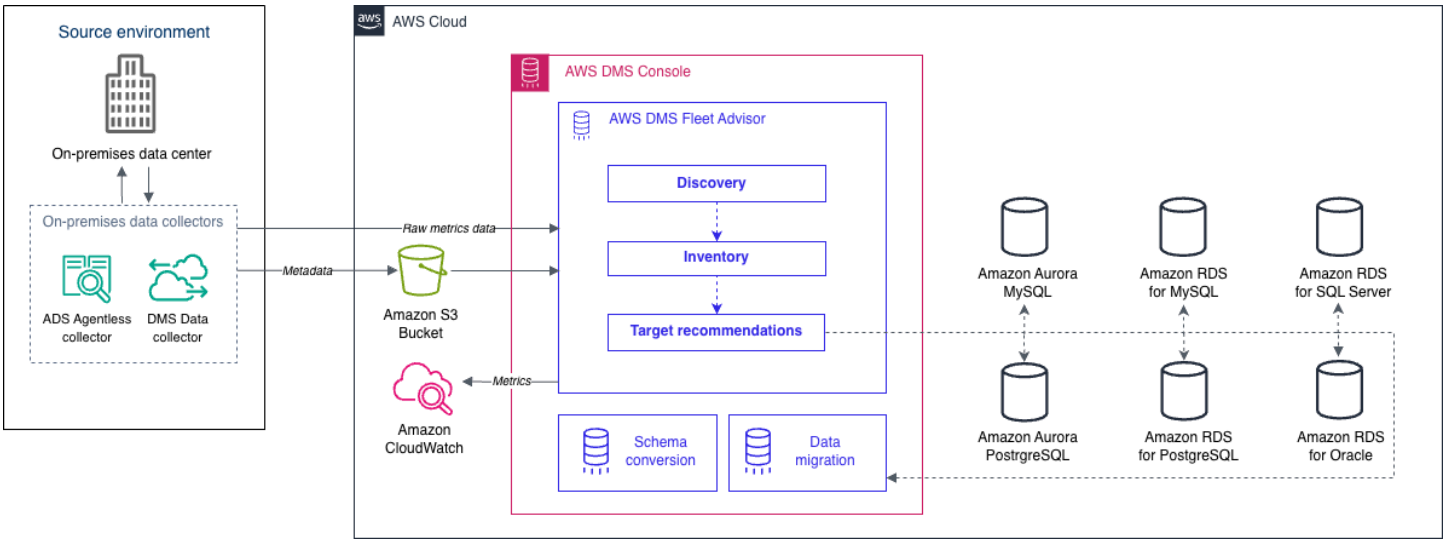
Sur la base des données découvertes sur le réseau, vous pouvez créer un inventaire afin de dresser la liste des serveurs de base de données en vue de poursuivre la collecte de données. Une fois qu'AWS DMS a collecté des informations relatives à vos serveurs, bases de données et schémas, vous pouvez analyser la faisabilité des migrations de bases de données que vous envisagez.

Pour les bases de données de votre inventaire que vous envisagez de migrer vers le AWS Cloud, DMS Fleet Advisor génère des recommandations dans le but de dimensionner correctement la cible. Pour générer des recommandations cibles, DMS Fleet Advisor prend en compte les métriques de votre collecteur de données et vos paramètres préférés. Une fois que DMS Fleet Advisor a généré des recommandations, vous pouvez consulter des informations détaillées pour chaque configuration de base de données cible. Les ingénieurs et administrateurs de bases de données de votre organisation peuvent utiliser les recommandations cibles de DMS Fleet Advisor pour planifier la migration de vos bases de données sur site vers AWS. Vous pouvez explorer les différentes options de migration disponibles et exporter ces recommandations AWS Pricing Calculator vers le pour optimiser davantage les coûts.

Pour obtenir la liste des bases de données sources prises en charge, consultez Sources pour DMS Fleet Advisor.

Pour obtenir la liste des bases de données utilisées par DMS Fleet Advisor pour générer des recommandations cibles, consultez Cibles pour DMS Fleet Advisor. DMS Fleet Advisor génère des recommandations similaires, par exemple, depuis la base de données Oracle source vers la base de données Oracle cible. DMS Fleet Advisor génère également des recommandations hétérogènes, telles que la migration depuis la source Oracle ou Microsoft SQL Server vers la base de données cible RDS pour PostgreSQL ou Aurora PostgreSQL.

Le schéma suivant illustre le processus d'élaboration des recommandations cibles d'AWS DMS Fleet Advisor.



Consultez les rubriques suivantes pour mieux comprendre comment utiliser AWS DMS Fleet Advisor.

Rubriques

- Régions AWS prises en charge
- Mise en route avec DMS Fleet Advisor
- Configuration d'AWS DMS Fleet Advisor
- Découverte des bases de données à migrer à l'aide de collecteurs de données
- Utilisation des inventaires à des fins d'analyse dans AWS DMS Fleet Advisor
- Utilisation de la fonctionnalité de recommandations cibles d'AWS DMS Fleet Advisor
- Limites de DMS Fleet Advisor

Régions AWS prises en charge

Vous pouvez utiliser DMS Fleet Advisor dans les Régions AWS suivantes.

Nom de la région	Région
US East (Virginie du Nord)	us-east-1
USA Est (Ohio)	us-east-2

Nom de la région	Région
USA Ouest (Californie du Nord)	us-west-1
US West (Oregon)	us-west-2
Asie-Pacifique (Hong Kong)	ap-east-1
Asie-Pacifique (Tokyo)	ap-northeast-1
Asie-Pacifique (Séoul)	ap-northeast-2
Asie-Pacifique (Osaka)	ap-northeast-3
Asie-Pacifique (Mumbai)	ap-south-1
Asie-Pacifique (Singapour)	ap-southeast-1
Asie-Pacifique (Sydney)	ap-southeast-2
Asie-Pacifique (Jakarta)	ap-southeast-3
Canada (Centre)	ca-central-1
Europe (Francfort)	eu-central-1
Europe (Stockholm)	eu-north-1
Europe (Irlande)	eu-west-1
Europe (Londres)	eu-west-2
Europe (Paris)	eu-west-3
Europe (Milan)	eu-south-3
Canada (Centre)	ca-central-1
Amérique du Sud (São Paulo)	sa-east-1

Nom de la région	Région
Moyen-Orient (Bahreïn)	me-south-1
Afrique (Le Cap)	af-south-1

Mise en route avec DMS Fleet Advisor

Vous pouvez utiliser DMS Fleet Advisor pour découvrir vos bases de données sources sur site à migrer vers le AWS Cloud. Vous pouvez ensuite déterminer la bonne cible de migration dans le AWS Cloud pour chacune de vos bases de données locales. Utilisez le flux de travail suivant pour créer un inventaire de vos bases de données sources et générer des recommandations cibles.

1. Créez un compartiment Amazon S3, des politiques IAM, des rôles et des utilisateurs. Pour de plus amples informations, veuillez consulter Création des ressources requises.
2. Créez des utilisateurs de base de données dotés des autorisations minimales requises pour le collecteur de données DMS. Pour de plus amples informations, veuillez consulter Création d'utilisateurs de base de données.
3. Créez et téléchargez un collecteur de données. Pour de plus amples informations, veuillez consulter Création d'un collecteur de données.
4. Installez le collecteur de données dans votre environnement local. Configurez ensuite votre collecteur de données pour vous assurer qu'il peut envoyer les données collectées à DMS Fleet Advisor. Pour de plus amples informations, veuillez consulter Installation d'un collecteur de données.
5. Découvrez les serveurs OS et de base de données figurant dans votre environnement de données. Pour de plus amples informations, veuillez consulter Découverte des serveurs de base de données et de système d'exploitation.
6. Collectez les métriques de métadonnées de base de données et d'utilisation des ressources. Pour de plus amples informations, veuillez consulter Collecte de données.
7. Analysez vos schémas et bases de données sources. DMS Fleet Advisor effectue une évaluation à grande échelle de vos bases de données afin d'identifier des schémas similaires. Pour de plus amples informations, veuillez consulter Utilisation des inventaires à des fins d'analyse dans AWS DMS Fleet Advisor.

8. Générez, visualisez et enregistrez une copie locale des recommandations cibles pour vos bases de données sources. Pour de plus amples informations, veuillez consulter Recommandations cibles.

Après avoir déterminé la cible de migration pour chaque base de données source, vous pouvez utiliser la conversion de schéma DMS pour convertir vos schémas de base de données vers une nouvelle plateforme. Ensuite, vous pouvez utiliser AWS DMS pour migrer les données. Pour plus d'informations, consultez Conversion de schémas de base de données à l'aide de la conversion de schéma DMS et Qu'est-ce que AWS Database Migration Service ?

Cette vidéo présente l'interface utilisateur de conversion de schéma DMS et vous aide à vous familiariser avec les principaux composants de ce service.

Configuration d'AWS DMS Fleet Advisor

Pour configurer AWS DMS Fleet Advisor, exécutez les tâches prérequises suivantes.

Rubriques

- Création des ressources AWS requises pour AWS DMS Fleet Advisor
- Création d'utilisateurs de base de données pour AWS DMS Fleet Advisor

Création des ressources AWS requises pour AWS DMS Fleet Advisor

DMS Fleet Advisor a besoin d'un ensemble de ressources AWS dans votre compte pour transférer et importer les informations d'inventaire, et pour mettre à jour le statut du collecteur de données DMS.

Avant de collecter des données et de créer des inventaires de bases de données et de schémas pour la première fois, vous devez remplir les conditions préalables suivantes.

Pour configurer votre compartiment Amazon S3 et vos ressources IAM, effectuez l'une des opérations suivantes :

- Configuration des ressources Amazon 3 et IAM à l'aide d'AWS CloudFormation (recommandé).
- Configuration des ressources Amazon 3 et IAM dans AWS Management Console

Configuration des ressources Amazon 3 et IAM à l'aide d'AWS CloudFormation

Une CloudFormation pile est un ensemble de ressources AWS que vous gérez comme une seule unité. Pour simplifier la création des ressources requises pour DMS Fleet Advisor, vous pouvez utiliser les fichiers AWS CloudFormation modèles pour créer des CloudFormation piles. Pour plus d'informations, consultez Création d'une pile via la console AWS CloudFormation dans le Guide de l'utilisateur AWS CloudFormation.

> **Note**
>
> Cette section concerne uniquement le collecteur autonome DMS Fleet Advisor. Pour plus d'informations sur l'utilisation d'un seul collecteur sur site pour collecter des informations à la fois sur les bases de données et les serveurs, consultez Collecteur sans agent d'Application Discovery Service dans le Guide de l'utilisateur AWS Application Discovery Service.

Ressources Amazon S3 et IAM créées par CloudFormation

Lorsque vous utilisez les CloudFormation modèles, ils créent des piles qui incluent les ressources suivantes dans votre Compte AWS :

- Un compartiment Amazon S3 nommé `dms-fleetadvisor-data-`*`accountId`*`-`*`region`*
- Un utilisateur IAM nommé `FleetAdvisorCollectorUser-`*`region`*
- Une fonction de service IAM nommée `FleetAdvisorS3Role-`*`region`*
- Une politique d'accès nommée `FleetAdvisorS3Role-`*`region`*`-Policy`
- Une politique d'accès nommée `FleetAdvisorCollectorUser-`*`region`*`-Policy`
- Un rôle lié à un service (SLR) IAM nommé `AWSServiceRoleForDMSFleetAdvisor`

Suivez les étapes ci-dessous pour configurer vos ressources avec CloudFormation.

- Étape 1 : Téléchargez les fichiers CloudFormation modèles
- Étape 2 : configurer Amazon S3 et IAM à l'aide de CloudFormation

Étape 1 : Téléchargez les fichiers CloudFormation modèles

Un CloudFormation modèle est une déclaration des AWS ressources qui constituent une pile. Le modèle est stocké sous la forme d'un fichier JSON.

Pour télécharger les fichiers CloudFormation modèles

1. Ouvrez le menu contextuel (clic droit) de l'un des liens suivants, puis choisissez Enregistrer le lien sous :

   - Si vous envisagez d'utiliser DMS Fleet Advisor, choisissez dms-fleetadvisor-iam-slr-s3.zip. Si vous avez déjà créé le SLR pour DMS Fleet Advisor, choisissez 3.zip dms-fleetadvisor-iam-s
   - Si vous envisagez d'utiliser le collecteur sans agent AWS Application Discovery Service (ADS) et que vous n'avez pas créé le SLR correspondant, choisissez dms-fleetadvisor-ads-iam -slr-s3.zip. Si vous avez déjà créé le SLR pour DMS Fleet Advisor avec ADS, choisissez dms-fleetadvisor-ads-iam -s3.zip.

2. Enregistrez le fichier sur votre ordinateur.

Étape 2 : configurer Amazon S3 et IAM à l'aide de CloudFormation

Lorsque vous utilisez le CloudFormation modèle pour IAM, il crée les ressources Amazon S3 et IAM répertoriées précédemment.

Pour configurer Amazon S3 et IAM à l'aide de CloudFormation

1. Ouvrez la CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.
2. Lancez l'assistant Créer une pile en sélectionnant Créer une pile et Avec de nouvelles ressources dans la liste déroulante.
3. Sur la page Créer une pile, procédez de la manière suivante :

   a. Pour Préparer le modèle, choisissez Le modèle est prêt.
   b. Pour Source du modèle, choisissez Charger un fichier de modèle.
   c. Pour Choisir un fichier, accédez à, puis choisissez dms-fleetadvisor-iam-slr-S3.json, -S3.json. dms-fleetadvisor-iam , dms-fleetadvisor-ads-iam-slr-s3.zip ou dms-fleetadvisor-ads-iam-s3.zip.
   d. Choisissez Suivant.

4. Sur la page Spécifier les détails de la pile, procédez comme suit :

   a. Dans le champ Nom de la pile, saisissez **`dms-fleetadvisor-iam-slr-s3`**, **`dms-fleetadvisor-iam-s3`**, **`dms-fleetadvisor-ads-iam-slr-s3`** ou **`dms-fleetadvisor-ads-iam-s3`**.
   b. Choisissez Next (Suivant).

5. Sur la page Configurer les options de pile, choisissez Suivant.
6. Sur la page Review dms-fleetadvisor-iam-slr -s3, Review dms-fleetadvisor-iam-s 3, Review dms-fleetadvisor-ads-iam -slr-s3 ou Review dms-fleetadvisor-ads-iam -s3, procédez comme suit :

   a. Cochez la case Je comprends qu'AWS CloudFormation peut créer des ressources IAM avec des noms personnalisés.

   b. Sélectionnez Envoyer.

   CloudFormation crée le compartiment S3 et les rôles et utilisateurs IAM requis par DMS Fleet Advisor. Dans le panneau de gauche, lorsque dms-fleetadvisor-iam-slr-s3, dms-fleetadvisor-iam-s3, dms-fleetadvisor-ads-iam-slr-s3 ou dms-fleetadvisor-ads-iam-s3 affiche CREATE_COMPLETE, passez à l'étape suivante.
7. Dans le panneau de gauche, choisissez dms-fleetadvisor-iam-slr-s3, dms-fleetadvisor-iam-s3, dms-fleetadvisor-ads-iam-slr-s3 ou -s3. dms-fleetadvisor-ads-iam Dans le volet de droite, procédez de la façon suivante :

   a. Choisissez Informations sur la pile. ***Votre stack possède un identifiant au format arn:aws:cloudformation : region : account-no:stack/ -s3/ identifier, arn:aws:cloudformation : region : account-no:stack/ 3/ identifier, arn:aws:cloudformation : region : account-no:stack/ -slr-s3/ dms-fleetadvisor-iam-slr identifier, ou arn:aws:cloudformation : region : account-no:stack/ -s3/ identifiant dms-fleetadvisor-iam-s dms-fleetadvisor-ads-iam dms-fleetadvisor-ads-iam*** .

   b. Sélectionnez Ressources. Vous devez voir ce qui suit :

      - Un compartiment Amazon S3 nommé `dms-fleetadvisor-data-`*`accountId`*`-`*`region`*
      - Un nom de rôle de service nommé `FleetAdvisorS3Role-`*`region`*
      - Un utilisateur IAM nommé `FleetAdvisorCollectorUser-`*`region`*
      - Un SLR IAM nommé `AWSServiceRoleForDMSFleetAdvisor` (si vous avez téléchargé `dms-fleet-advisor-iam-slr-s3.zip` oudms-fleet-advisor-ads-iam-slr-s3.zip).
      - Une politique d'accès nommée `FleetAdvisorS3Role-`*`region`*`-Policy`
      - Une politique d'accès nommée `FleetAdvisorCollectorUser-`*`region`*`-Policy`

Configuration des ressources Amazon 3 et IAM dans AWS Management Console

Créer un compartiment Amazon S3

Créez un compartiment Amazon S3 afin d'y stocker les métadonnées d'inventaire. Nous vous recommandons de préconfigurer ce compartiment S3 avant d'utiliser DMS Fleet Advisor. AWS DMS stocke vos métadonnées d'inventaire DMS Fleet Advisor dans ce compartiment S3.

Pour plus d'informations sur la création d'un compartiment Amazon S3, consultez Créer votre premier compartiment S3 dans le Guide de l'utilisateur Amazon S3.

> **Note**
>
> DMS Fleet Advisor ne prend en charge que les compartiments chiffrés SSE-S3.

Pour créer un compartiment Amazon S3 afin d'y stocker des informations sur l'environnement de données local

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Choisissez Créer un compartiment.
3. Sur la page Créer un compartiment, entrez un nom unique global incluant votre nom de connexion pour le compartiment, tel que fa-bucket-***yoursignin***.
4. Choisissez la Région AWS dans laquelle vous utilisez DMS Fleet Advisor.
5. Conservez les autres paramètres et choisissez Créer un compartiment.

Création de ressources IAM

Dans cette section, vous allez créer des ressources IAM pour votre collecteur de données, l'utilisateur IAM et DMS Fleet Advisor.

Rubriques

- Création de ressources IAM pour votre collecteur de données
- Création du rôle lié au service DMS Fleet Advisor

Création de ressources IAM pour votre collecteur de données

Pour vous assurer que votre collecteur de données fonctionne correctement et charge les métadonnées collectées dans votre compartiment Amazon S3, créez les politiques suivantes. Ensuite, créez un utilisateur IAM avec les autorisations minimales suivantes. Pour plus d'informations sur le collecteur de données DMS, consultez Découverte des bases de données à migrer à l'aide de collecteurs de données.

Pour créer une politique IAM permettant à DMS Fleet Advisor et à votre collecteur de données d'accéder à Amazon S3

1. Connectez-vous à l'AWS Management Console et ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Politiques.
3. Sélectionnez Create policy (Créer une politique).
4. Sur la page Créer une politique, choisissez l'onglet JSON.
5. Collez le code JSON suivant dans l'éditeur en remplaçant l'exemple de code. Remplacez *fa_bucket* par le nom du compartiment Amazon S3 que vous avez créé dans la section précédente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject*",
                "s3:GetBucket*",
                "s3:List*",
                "s3:DeleteObject*",
                "s3:PutObject*"
            ],
            "Resource": [
                "arn:aws:s3:::fa_bucket",
                "arn:aws:s3:::fa_bucket/*"
            ]
        }
    ]
}
```

6. Choisissez Next: Tags (Suivant : Balises), puis Next: Review (Suivant : Vérification).
7. Entrez **`FleetAdvisorS3Policy`** pour Nom*, puis choisissez Créer une politique.

Pour créer une politique IAM permettant au collecteur de données DMS d'accéder à DMS Fleet Advisor

1. Connectez-vous à l'AWS Management Console et ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Politiques.
3. Sélectionnez Create policy (Créer une politique).
4. Sur la page Créer une politique, choisissez l'onglet JSON.
5. Collez le code JSON suivant dans l'éditeur en remplaçant l'exemple de code.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dms:DescribeFleetAdvisorCollectors",
                "dms:ModifyFleetAdvisorCollectorStatuses",
                "dms:UploadFileMetadataList"
            ],
            "Resource": "*"
        }
    ]
}
```

6. Choisissez Next: Tags (Suivant : Balises), puis Next: Review (Suivant : Vérification).
7. Entrez **`DMSCollectorPolicy`** pour Nom*, puis choisissez Créer une politique.

Pour créer un utilisateur IAM avec des autorisations minimales afin d'utiliser le collecteur de données DMS

1. Connectez-vous à la AWS Management Console et ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez utilisateurs.
3. Sélectionnez Ajouter des utilisateurs.

4. Sur la page Ajouter un utilisateur, entrez **`FleetAdvisorCollectorUser`** pour Nom d'utilisateur*. Choisissez Clé d'accès - Accès par programmation pour Sélectionner le type d'accès AWS. Sélectionnez Next: Permissions (Étape suivante : autorisations).
5. Dans la section Définir les autorisations, choisissez Attacher directement les politiques existantes.
6. Utilisez le contrôle de recherche pour rechercher et sélectionner les politiques DMS CollectorPolicy et FleetAdvisorS3Policy que vous avez créées auparavant. Choisissez Suivant : Balises.
7. Sur la page Tags (Identifications), choisissez Next: Review (Suivant : Révision).
8. Sur la page Review (Vérification), choisissez Create user (Créer un utilisateur). Sur la page suivante, choisissez Télécharger .csv pour enregistrer les nouvelles informations d'identification utilisateur. Utilisez ces informations d'identification avec DMS Fleet Advisor pour obtenir les autorisations d'accès minimales requises.

Pour créer un rôle IAM permettant à DMS Fleet Advisor et à votre collecteur de données d'accéder à Amazon S3

1. Connectez-vous à la AWS Management Console, puis ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Roles (Rôles).
3. Sélectionnez Create role (Créer un rôle).
4. Sur la page Sélectionner une entité de confiance, pour Type d'entité approuvée, choisissez Service AWS. Pour Cas d'utilisation d'autres services AWS, choisissez DMS.
5. Cochez la case DMS et choisissez Suivant.
6. Sur la page Ajouter des autorisations, choisissez FleetAdvisorS3Policy. Choisissez Suivant.
7. Sur la page Nommer, vérifier et créer, entrez **`FleetAdvisorS3Role`** pour Nom du rôle, puis choisissez Créer un rôle.
8. Sur la page Rôles, entrez **`FleetAdvisorS3Role`** pour Nom du rôle. Choisissez FleetAdvisorS3Role.
9. Sur la page FleetAdvisorS3Role, choisissez l'onglet Relations de confiance. Choisissez Edit trust policy (Modifier la politique d'approbation).
10. Sur la page Modifier la politique d'approbation, collez le code JSON suivant dans l'éditeur en remplaçant le texte existant.

```
{
 "Version": "2012-10-17",
 "Statement": [
    {
     "Sid": "",
     "Effect": "Allow",
     "Principal": {
       "Service": [
         "dms.amazonaws.com",
         "dms-fleet-advisor.amazonaws.com"
        ]
      },
     "Action": "sts:AssumeRole"
    }
  ]
}
```

La politique précédente accorde l'autorisation `sts:AssumeRole` aux services utilisés par AWS DMS pour importer les données collectées depuis le compartiment Amazon S3.

11. Choisissez Mettre à jour une politique.

Création du rôle lié au service DMS Fleet Advisor

DMS Fleet Advisor utilise un rôle lié à un service pour gérer les CloudWatch métriques Amazon dans votre. Compte AWS DMS Fleet Advisor utilise ce rôle lié au service pour publier en votre nom les indicateurs de performance de la CloudWatch base de données collectés.

Pour créer le rôle lié à un service pour DMS Fleet Advisor

1. Connectez-vous à la AWS Management Console, puis ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, sélectionnez Rôles. Puis, choisissez Créer un rôle.
3. Pour Type d'entité de confiance, choisissez Service AWS.
4. Pour Cas d'utilisation pour d'autres services AWS, choisissez DMS – Fleet Advisor.
5. Cochez la case DMS – Fleet Advisor et choisissez Suivant.
6. Sur la page Ajouter des autorisations, sélectionnez Suivant.
7. Sur la page Nommer, vérifier et créer, choisissez Créer un rôle.

Vous pouvez également créer ce rôle lié à un service à partir de l'API AWS ou de l'interface de ligne de commande AWS. Pour plus d'informations, consultez Création d'un rôle lié à un service pour AWS DMS Fleet Advisor.

Après avoir créé le rôle lié à un service pour DMS Fleet Advisor, vous pouvez consulter les métriques de performance de vos bases de données sources dans les recommandations cibles. Vous pouvez également consulter ces statistiques et les consulter dans votre CloudWatch compte. Pour plus d'informations, consultez Recommandations cibles.

Pour créer une politique IAM requise pour le rôle lié au service DMS Fleet Advisor

Les autorisations minimales nécessaires pour créer le rôle lié à un service sont spécifiées dans la politique `DMSFleetAdvisorCreateServiceLinkedRolePolicy`. Créez cette politique IAM pour votre compte si vous ne parvenez pas à créer le rôle lié à un service.

1. Connectez-vous à l'AWS Management Console et ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Politiques.
3. Sélectionnez Create policy (Créer une politique).
4. Sur la page Créer une politique, choisissez l'onglet JSON.
5. Collez le code JSON suivant dans l'éditeur en remplaçant l'exemple de code.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/dms-fleet-
advisor.amazonaws.com/AWSServiceRoleForDMSFleetAdvisor*",
            "Condition": {"StringLike": {"iam:AWSServiceName": "dms-fleet-
advisor.amazonaws.com"}}
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
```

```
            "Resource": "arn:aws:iam::*:role/aws-service-role/dms-fleet-
advisor.amazonaws.com/AWSServiceRoleForDMSFleetAdvisor*"
        }
    ]
}
```

6. Choisissez Next: Tags (Suivant : Balises), puis Next: Review (Suivant : Vérification).
7. Entrez **`DMSFleetAdvisorCreateServiceLinkedRolePolicy`** pour Nom*, puis choisissez Créer une politique.

Vous pouvez désormais utiliser cette politique pour créer le rôle lié à un service pour DMS Fleet Advisor.

Création d'utilisateurs de base de données pour AWS DMS Fleet Advisor

Cette section décrit la manière de créer des utilisateurs pour vos bases de données sources avec les autorisations minimales requises pour le collecteur de données DMS.

Cette section contient les rubriques suivantes :

- Utilisation d'un utilisateur de base de données avec AWS DMS Fleet Advisor
- Création d'un utilisateur de base de données avec MySQL
- Création d'un utilisateur de base de données avec Oracle
- Création d'un utilisateur de base de données avec PostgreSQL
- Création d'un utilisateur de base de données avec Microsoft SQL Server
- Suppression d'utilisateurs de base de données

Utilisation d'un utilisateur de base de données avec AWS DMS Fleet Advisor

Vous pouvez utiliser un utilisateur de base de données autre que `root` avec le collecteur de données DMS. Spécifiez le nom d'utilisateur et le mot de passe après avoir ajouté la base de données à l'inventaire, mais avant d'exécuter votre collecteur de données. Pour plus d'informations sur l'ajout de bases de données à l'inventaire, consultez Gestion des objets surveillés.

Une fois que vous avez fini d'utiliser le collecteur de données DMS, vous pouvez supprimer les utilisateurs de base de données que vous avez créés. Pour plus d'informations, consultez Suppression d'utilisateurs de base de données.

> **Important**
>
> Dans les exemples suivants, remplacez *{your_user_name}* par le nom de l'utilisateur de base de données que vous avez créé pour la base de données. Remplacez ensuite *{your_password}* par un mot de passe sécurisé.

Création d'un utilisateur de base de données avec MySQL

Pour créer un utilisateur de base de données dans une base de données source MySQL, utilisez le script suivant. Assurez-vous de conserver une seule version de l'instruction GRANT qui dépend de la version de la base de données MySQL.

```
CREATE USER {your_user_name} identified BY '{your_password}';

GRANT PROCESS ON *.* TO {your_user_name};
GRANT REFERENCES ON *.* TO {your_user_name};
GRANT TRIGGER ON *.* TO {your_user_name};
GRANT EXECUTE ON *.* TO {your_user_name};

# For MySQL versions lower than 8.0, use the following statement.
GRANT SELECT, CREATE TEMPORARY TABLES ON `temp`.* TO {your_user_name};

# For MySQL versions 8.0 and higher, use the following statement.
GRANT SELECT, CREATE TEMPORARY TABLES ON `mysql`.* TO {your_user_name};

GRANT SELECT ON performance_schema.* TO {your_user_name};

SELECT
    IF(round(Value1 + Value2 / 100 + Value3 / 10000, 4) > 5.0129, 'GRANT EVENT ON *.*
 TO {your_user_name};', 'SELECT ''Events are not applicable'';') sql_statement
INTO @stringStatement
FROM (
       SELECT
         substring_index(ver, '.', 1)                              value1,
         substring_index(substring_index(ver, '.', 2), '.', - 1) value2,
         substring_index(ver, '.', - 1)                            value3
       FROM  (
         SELECT
           IF((@@version regexp '[^0-9\.]+') != 0, @@innodb_version, @@version) AS ver
         FROM dual
       ) vercase
```

```
     ) v;

PREPARE sqlStatement FROM @stringStatement;
SET @stringStatement := NULL;
EXECUTE sqlStatement;
DEALLOCATE PREPARE sqlStatement;
```

Création d'un utilisateur de base de données avec Oracle

Pour créer un utilisateur de base de données dans une base de données source Oracle, utilisez le script suivant.

Pour exécuter ce script SQL, connectez-vous à la base de données Oracle à l'aide de privilèges SYSDBA. Après avoir exécuté ce script SQL, connectez-vous à la base de données à l'aide des informations d'identification de l'utilisateur que vous avez créé avec ce script. Utilisez également les informations d'identification de cet utilisateur pour exécuter le collecteur de données DMS.

Le script suivant ajoute le préfixe C## au nom de l'utilisateur pour les bases de données de conteneurs (CDB) multilocataires Oracle.

```
CREATE USER {your_user_name} IDENTIFIED BY "{your_password}";
GRANT CREATE SESSION TO {your_user_name};
GRANT SELECT ANY DICTIONARY TO {your_user_name};
GRANT SELECT ON DBA_WM_SYS_PRIVS TO {your_user_name};
BEGIN
    DBMS_NETWORK_ACL_ADMIN.CREATE_ACL(
        acl => UPPER('{your_user_name}') || '_Connect_Access.xml',
        description => 'Connect Network',
        principal => UPPER('{your_user_name}'),
        is_grant => TRUE,
        privilege => 'resolve',
        start_date => NULL,
        end_date => NULL);

    DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL(
        acl => UPPER('{your_user_name}') || '_Connect_Access.xml',
        host => '*',
        lower_port => NULL,
        upper_port => NULL);
END;
```

Création d'un utilisateur de base de données avec PostgreSQL

Pour créer un utilisateur de base de données dans une base de données source PostgreSQL, utilisez le script suivant.

```
CREATE USER "{your_user_name}" WITH LOGIN PASSWORD '{your_password}';
GRANT pg_read_all_settings TO "{your_user_name}";

-- For PostgreSQL versions 10 and higher, add the following statement.
GRANT EXECUTE ON FUNCTION pg_ls_waldir() TO "{your_user_name}";
```

Création d'un utilisateur de base de données avec Microsoft SQL Server

Pour créer un utilisateur de base de données dans une base de données source Microsoft SQL Server, utilisez le script suivant.

```
USE master
GO

IF NOT EXISTS (SELECT * FROM sys.sql_logins WHERE name = N'{your_user_name}')

 CREATE LOGIN [{your_user_name}] WITH PASSWORD=N'{your_password}',
 DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF,
 CHECK_POLICY=OFF

GO

GRANT VIEW SERVER STATE TO [{your_user_name}]

GRANT VIEW ANY DEFINITION TO [{your_user_name}]

GRANT VIEW ANY DATABASE TO [{your_user_name}]

IF LEFT(CONVERT(SYSNAME,SERVERPROPERTY('ProductVersion')), CHARINDEX('.',
 CONVERT(SYSNAME,SERVERPROPERTY('ProductVersion')), 0)-1) >= 12
 EXECUTE('GRANT CONNECT ANY DATABASE TO [{your_user_name}]')

DECLARE @dbname VARCHAR(100)
DECLARE @statement NVARCHAR(max)

DECLARE db_cursor CURSOR
LOCAL FAST_FORWARD
FOR
```

```
  SELECT
    name
  FROM MASTER.sys.databases
  WHERE state = 0
    AND is_read_only = 0
        OPEN db_cursor
FETCH NEXT FROM db_cursor INTO @dbname
    WHILE @@FETCH_STATUS = 0
BEGIN

  SELECT @statement = 'USE '+ quotename(@dbname) +';'+ '
    IF NOT EXISTS (SELECT * FROM sys.syslogins WHERE name = ''{your_user_name}'') OR
 NOT EXISTS (SELECT * FROM sys.sysusers WHERE name = ''{your_user_name}'')
   CREATE USER [{your_user_name}] FOR LOGIN [{your_user_name}];

  EXECUTE sp_addrolemember N''db_datareader'', [{your_user_name}]'

  BEGIN TRY
    EXECUTE sp_executesql @statement
  END TRY
  BEGIN CATCH
    DECLARE @err NVARCHAR(255)

    SET @err = error_message()

    PRINT @dbname
    PRINT @err
  END CATCH

  FETCH NEXT FROM db_cursor INTO @dbname
END
CLOSE db_cursor
  DEALLOCATE db_cursor

USE msdb
GO

GRANT EXECUTE ON dbo.agent_datetime TO [{your_user_name}]
```

Suppression d'utilisateurs de base de données

Une fois toutes les tâches de collecte de données terminées, vous pouvez supprimer les utilisateurs de base de données que vous avez créés pour le collecteur de données DMS. Vous pouvez utiliser

les scripts suivants pour supprimer de vos bases de données les utilisateurs dotés d'autorisations minimales.

Pour supprimer l'utilisateur de la base de données MySQL, exécutez le script suivant.

```
DROP USER IF EXISTS "{your_user_name}";
```

Pour supprimer l'utilisateur de la base de données Oracle, exécutez le script suivant.

```
DECLARE
  -- Input parameters, please set correct value
  cnst$user_name CONSTANT VARCHAR2(255) DEFAULT '{your_user_name}';

  -- System variables, please, don't change
  var$is_exists INTEGER DEFAULT 0;
BEGIN
  SELECT COUNT(hal.acl) INTO var$is_exists
  FROM dba_host_acls hal
  WHERE hal.acl LIKE '%' || UPPER(cnst$user_name) || '_Connect_Access.xml';
  IF var$is_exists > 0 THEN
    DBMS_NETWORK_ACL_ADMIN.DROP_ACL(
      acl => UPPER(cnst$user_name) || '_Connect_Access.xml');
  END IF;
  SELECT COUNT(usr.username) INTO var$is_exists
  FROM all_users usr
  WHERE usr.username = UPPER(cnst$user_name);
  IF var$is_exists > 0 THEN
    EXECUTE IMMEDIATE 'DROP USER ' || cnst$user_name || ' CASCADE';
  END IF;
END;
```

Pour supprimer l'utilisateur de la base de données PostgreSQL, exécutez le script suivant.

```
DROP USER IF EXISTS "{your_user_name}";
```

Pour supprimer l'utilisateur de la base de données SQL Server, exécutez le script suivant.

```
USE msdb
GO

REVOKE EXECUTE ON dbo.agent_datetime TO [{your_user_name}]
```

```
USE master
GO

DECLARE @dbname VARCHAR(100)
DECLARE @statement NVARCHAR(max)

DECLARE db_cursor CURSOR
LOCAL FAST_FORWARD
FOR
SELECT
  name
FROM MASTER.sys.databases
WHERE state = 0
  AND is_read_only = 0
    OPEN db_cursor
FETCH NEXT FROM db_cursor INTO @dbname
  WHILE @@FETCH_STATUS = 0
BEGIN

SELECT @statement = 'USE '+ quotename(@dbname) +';'+ '
  EXECUTE sp_droprolemember N''db_datareader'', [{your_user_name}]

  IF EXISTS (SELECT * FROM sys.syslogins WHERE name = ''{your_user_name}'')
    OR EXISTS (SELECT * FROM sys.sysusers WHERE name = ''{your_user_name}'')
    DROP USER [{your_user_name}];'

BEGIN TRY
EXECUTE sp_executesql @statement
END TRY
BEGIN CATCH
  DECLARE @err NVARCHAR(255)

  SET @err = error_message()

  PRINT @dbname
  PRINT @err
END CATCH

FETCH NEXT FROM db_cursor INTO @dbname
END
CLOSE db_cursor
  DEALLOCATE db_cursor

GO
```

```
IF EXISTS (SELECT * FROM sys.sql_logins WHERE name = N'{your_user_name}')
  DROP LOGIN [{your_user_name}]  -- Use for SQL login

GO
```

Découverte des bases de données à migrer à l'aide de collecteurs de données

Pour découvrir votre infrastructure de données source, vous pouvez utiliser le collecteur sans agent d'AWSApplication Discovery Service ou des collecteurs de données AWS DMS. Le collecteur sans agent d'ADS est une application sur site qui collecte des informations sur votre environnement local à l'aide de méthodes sans agent, notamment des informations sur le profil du serveur (système d'exploitation, nombre de processeurs, quantité de mémoire RAM, par exemple), des métadonnées de bases de données et des métriques d'utilisation. Installez le collecteur sans agent en tant que machine virtuelle (VM) dans votre environnement VMware vCenter Server à l'aide d'un fichier Open Virtualization Archive (OVA). Un collecteur de AWS DMS données est une application Windows que vous installez dans votre environnement local. Cette application se connecte à votre environnement de données et collecte des métadonnées et des métriques de performance à partir de vos serveurs d'analytique et de base de données sur site. Une fois que les métadonnées de bases de données et les métriques de performance ont été collectées par le collecteur sans agent d'ADS ou un collecteur de données DMS, DMS Fleet Advisor dresse un inventaire des serveurs, des bases de données et des schémas que vous pouvez migrer vers AWS Cloud.

Le collecteur de données DMS est une application Windows qui utilise des bibliothèques, des connecteurs et des fournisseurs de données .NET pour se connecter à vos bases de données sources à des fins de découverte de bases de données et de collecte de données.

Le collecteur de données DMS s'exécute sous Windows. Toutefois, votre collecteur de données DMS peut collecter des données auprès de tous les fournisseurs de bases de données pris en charge, quel que soit le serveur OS sur lequel ils s'exécutent.

Le collecteur de données DMS utilise un protocole RTPS protégé avec chiffrement TLS pour établir une connexion sécurisée avec DMS Fleet Advisor. Par conséquent, vos données sont chiffrées pendant le transit par défaut.

AWS DMS a le nombre maximum de collecteurs de données que vous pouvez créer pour votre Compte AWS. Pour en savoir plus sur les quotas de service AWS DMS, consultez Quotas pour AWS Database Migration Service.

Rubriques

- Autorisations pour un collecteur de données DMS
- Création d'un collecteur de données pour AWS DMS Fleet Advisor
- Installation et configuration d'un collecteur de données
- Découverte des serveurs de base de données et de système d'exploitation à surveiller
- Gestion des objets surveillés
- Utilisation du protocole SSL avec AWS DMS Fleet Advisor
- Collecte de données pour AWS DMS Fleet Advisor
- Résolution des problèmes liés au collecteur de données DMS

Autorisations pour un collecteur de données DMS

Les utilisateurs de base de données que vous créez pour le collecteur de données DMS doivent disposer d'autorisations de lecture. Toutefois, dans certains cas, l'utilisateur de la base de données a besoin de l'autorisation `EXECUTE`. Pour plus d'informations, consultez Création d'utilisateurs de base de données pour AWS DMS Fleet Advisor.

Le collecteur de données DMS nécessite des autorisations supplémentaires pour exécuter les scripts de découverte.

- Pour la découverte du système d'exploitation, le collecteur de données DMS a besoin d'informations d'identification pour que le serveur de domaine puisse exécuter les demandes à l'aide du protocole LDAP.
- Pour la découverte de bases de données dans Linux, le collecteur de données DMS a besoin d'informations d'identification avec octrois `sudo` `SSH`. Vous devez également configurer vos serveurs Linux pour autoriser l'exécution de scripts SSH distants.
- Pour la découverte de bases de données dans Windows, le collecteur de données DMS a besoin d'informations d'identification avec octrois pour exécuter des requêtes Windows Management Instrumentation (WMI) et WMI Query Language (WQL) et lire le registre. Vous devez également configurer vos serveurs Windows pour autoriser l'exécution à distance de WMI, de WQL et PowerShell de scripts.

Création d'un collecteur de données pour AWS DMS Fleet Advisor

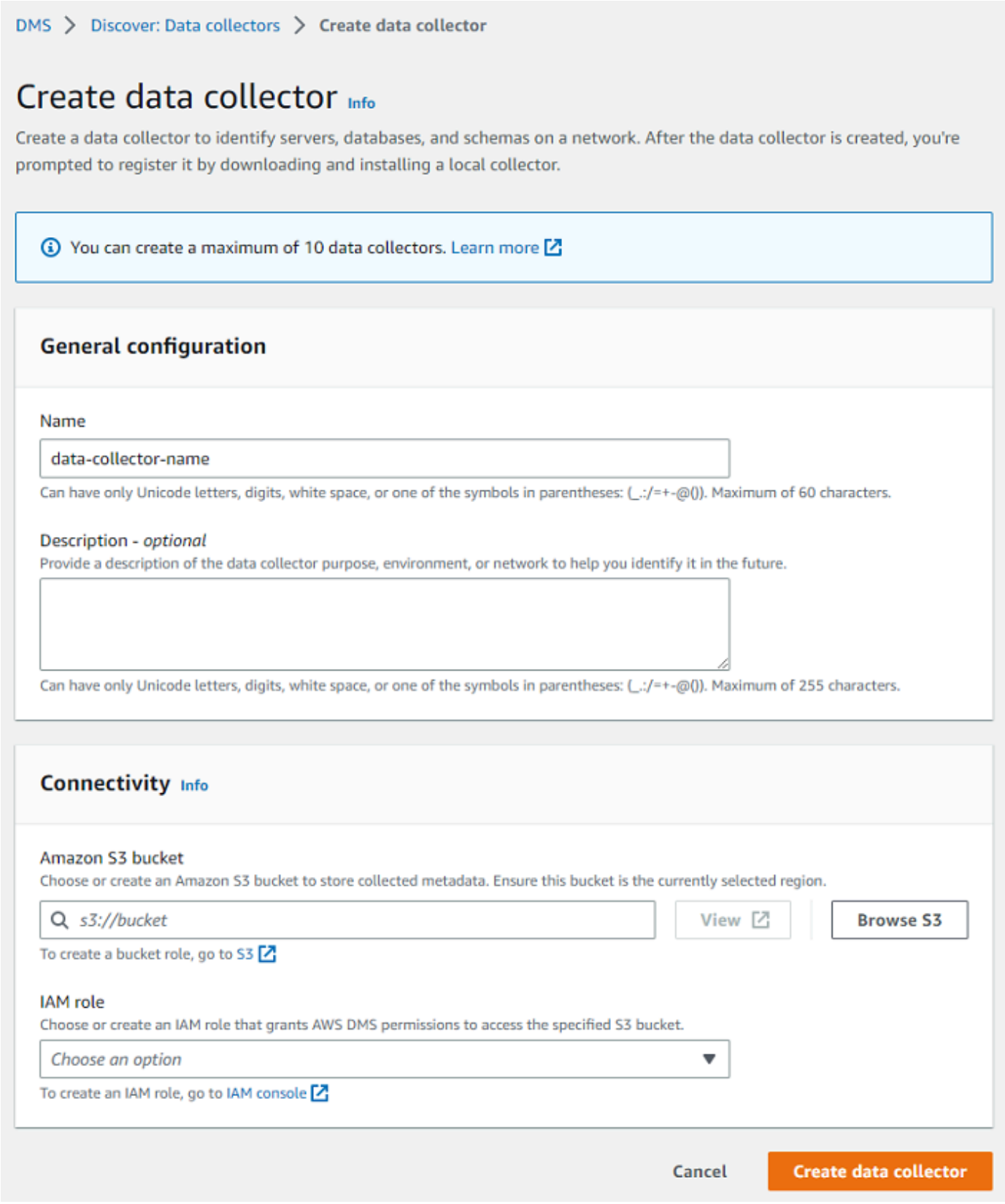
Découvrez comment créer et télécharger un collecteur de données DMS.

Avant de créer un collecteur de données, utilisez la console IAM pour créer un rôle lié à un service pour DMS Fleet Advisor. Ce rôle permet aux principaux de publier des points de données métriques sur Amazon CloudWatch. DMS Fleet Advisor utilise ce rôle pour afficher des graphiques avec des métriques de base de données. Pour plus d'informations, consultez Création d'un rôle lié à un service pour AWS DMS Fleet Advisor.

Pour créer et télécharger un collecteur de données

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.

   Choisissez la région dans laquelle vous utilisez DMS Fleet Advisor.
2. Dans le volet de navigation, choisissez Collecteurs de données sous Découvrir. La page Collecteurs de données s'ouvre.
3. Choisissez Créer un collecteur de données. La page Créer un collecteur de données s'ouvre.



4. Pour Nom dans la section Configuration générale, entrez le nom de votre collecteur de données.

5. Dans la section Connectivité, choisissez Parcourir S3. Choisissez le compartiment Amazon S3 que vous avez préconfiguré dans la liste qui s'affiche.

   AWS DMS stocke vos métadonnées d'inventaire DMS Fleet Advisor dans ce compartiment S3. Veillez à ce que votre compartiment Amazon S3 se trouve dans la Région AWS où AWS DMS Fleet Advisor est actuellement en cours d'exécution.

   > **Note**
   >
   > DMS Fleet Advisor ne prend en charge que les compartiments chiffrés SSE-S3.

6. Dans la liste des rôles IAM, choisissez le rôle IAM que vous avez préconfiguré dans la liste qui s'affiche. Ce rôle accorde à AWS DMS des autorisations d'accès au compartiment Amazon S3 spécifié.
7. Choisissez Créer un collecteur de données. La page Collecteurs de données s'ouvre et le collecteur de données créé apparaît dans la liste.

   Lorsque vous créez votre premier collecteur de données, AWS DMS configure un environnement dans votre compartiment Amazon S3 qui formate les données et stocke les attributs à utiliser avec DMS Fleet Advisor.
8. Choisissez Télécharger le collecteur local dans la bannière d'informations pour télécharger le collecteur de données que vous venez de créer. Un message vous informe que le téléchargement est en cours. Une fois le téléchargement terminé, vous pouvez accéder au fichier `AWS_DMS_Collector_Installer_`*`version_number`*`.msi`.

Vous pouvez désormais installer le collecteur de données DMS sur votre client. Pour plus d'informations, consultez Installation et configuration d'un collecteur de données.

Installation et configuration d'un collecteur de données

Découvrez comment installer votre collecteur de données DMS, comment spécifier les informations d'identification pour le transfert de données et comment ajouter un serveur LDAP à votre projet.

Le tableau suivant décrit la configuration matérielle et logicielle requise pour l'installation d'un collecteur de données DMS.

Minimum	Recommandée
Processeur : 2 cœurs avec un score de référence de CPU supérieur à 8 000	Processeur : 4 cœurs avec un score de référence de CPU supérieur à 11 000
RAM : 8 Go	RAM : 16 Go
Taille du disque dur : 256 Go	Taille du disque dur : 512 Go
Système d'exploitation : Microsoft Windows Server 2012 ou version ultérieure	Système d'exploitation : Windows Server 2016 ou version ultérieure

Pour installer un collecteur de données sur un client de votre réseau

1. Exécutez le programme d'installation .MSI. La page de l'Assistant de configuration du collecteur AWS DMS Fleet Advisor apparaît.
2. Choisissez Suivant. Le contrat de licence d'utilisateur final apparaît.
3. Lisez et acceptez le contrat de licence d'utilisateur final.
4. Choisissez Suivant. La page Dossier de destination apparaît.
5. Choisissez Suivant pour installer le collecteur de données dans le répertoire par défaut.

   Ou choisissez Modifier pour accéder à un autre répertoire d'installation. Ensuite, sélectionnez Suivant.
6. Sur la page Raccourci Bureau, cochez la case pour installer une icône sur votre bureau.
7. Choisissez Installer. Le collecteur de données est installé dans le répertoire que vous avez choisi.
8. Sur la page Assistant de configuration du collecteur DMS terminé, choisissez Lancer le collecteur AWS DMS, puis choisissez Terminer.

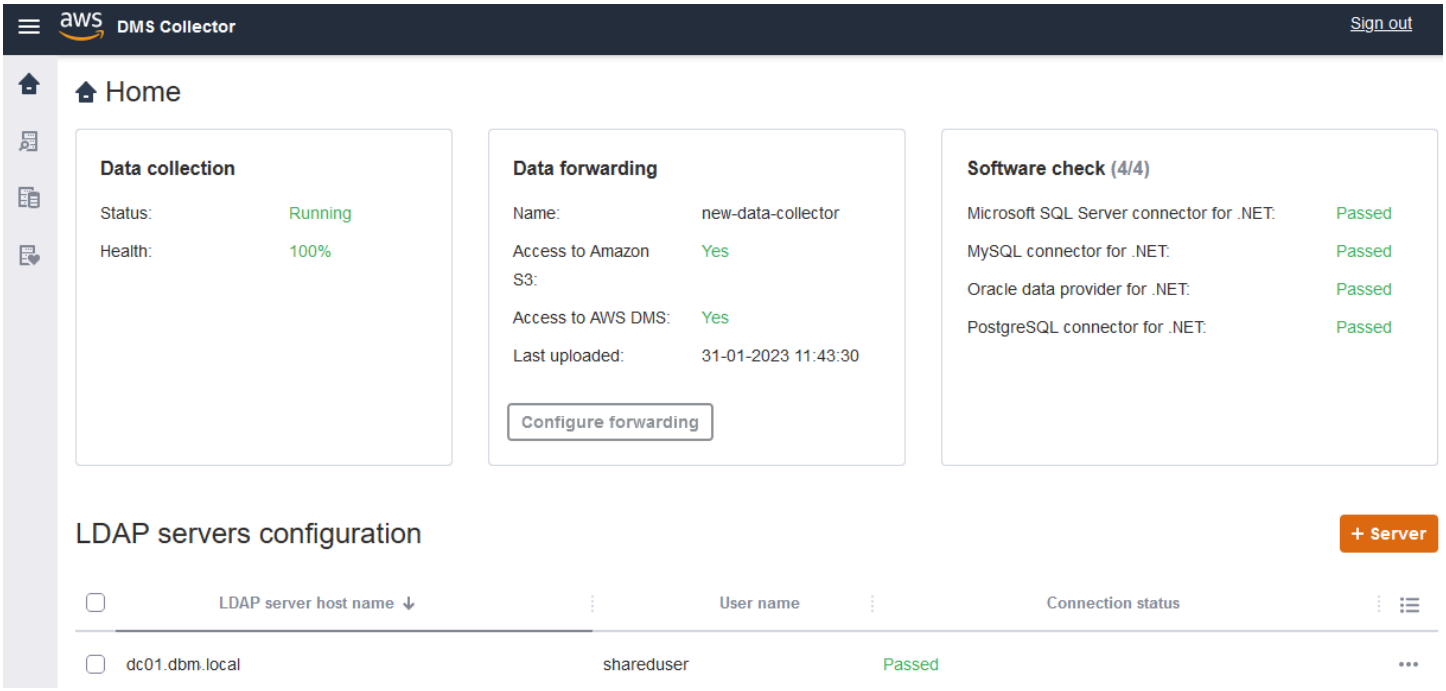
Votre collecteur de données DMS utilise des bibliothèques .NET, des connecteurs et des fournisseurs de données pour se connecter à vos bases de données sources. Le programme d'installation du collecteur de données DMS installe automatiquement ce logiciel requis pour toutes les bases de données prises en charge sur votre serveur.

Après avoir installé le collecteur de données, vous pouvez l'exécuter à partir d'un navigateur en entrant **`http://localhost:11000/`** comme adresse. Éventuellement, dans le menu Démarrer de

Microsoft Windows, choisissez Collecteur AWS DMS dans la liste des programmes. Lorsque vous exécutez le collecteur de données DMS pour la première fois, il vous est demandé de configurer des informations d'identification. Créez le nom d'utilisateur et le mot de passe pour vous connecter au collecteur de données.

Sur la page d'accueil du collecteur de données DMS, vous trouverez des informations relatives à la préparation et à l'exécution de la collecte de métadonnées, notamment les conditions de statut suivantes :

- Statut et état de votre collecte de données.
- Accessibilité à votre compartiment Amazon S3 et à AWS DMS, afin que le collecteur de données puisse transférer des données vers AWS DMS.
- Connectivité à vos pilotes de base de données installés.
- Informations d'identification d'un serveur LDAP pour effectuer la découverte initiale.



Le collecteur de données DMS utilise un annuaire LDAP pour recueillir des informations sur les machines et les serveurs de base de données de votre réseau. Le protocole LDAP (Lightweight Directory Protocol) est un protocole d'application standard ouvert. Il est utilisé pour accéder à des services d'information d'annuaire distribués et les gérer via un réseau IP. Vous pouvez ajouter un serveur LDAP existant à votre projet de collecteur de données que vous pouvez utiliser pour découvrir des informations sur l'infrastructure de vos systèmes. Pour ce faire, choisissez l'option

+Serveur, puis spécifiez un nom de domaine complet (FQDN) et les informations d'identification de votre contrôleur de domaine. Après avoir ajouté le serveur, validez la vérification de la connexion. Pour commencer avec le processus de découverte, consultez Découverte des serveurs de base de données et de système d'exploitation à surveiller.

Configuration des informations d'identification pour le transfert de données

Après avoir installé le collecteur de données, assurez-vous que cette application peut envoyer les données collectées à AWS DMS Fleet Advisor.

Pour configurer les informations d'identification pour le transfert de données dans AWS DMS Fleet Advisor

1. Sur la page d'accueil du collecteur de données DMS, dans la section Transfert de données, choisissez Configurer le transfert. La boîte de dialogue Configurer les informations d'identification pour le transfert de données s'ouvre.
2. Choisissez la Région AWS dans laquelle vous envisagez d'utiliser DMS Fleet Advisor.
3. Entrez votre ID de clé d'accès AWS et votre clé d'accès secrète AWS obtenus précédemment lors de la création des ressources IAM. Pour plus d'informations, consultez Création de ressources IAM.
4. Choisissez Parcourir les collecteurs de données.

   Si vous n'avez pas encore créé de collecteur de données dans la région spécifiée, créez-en un avant de continuer. Pour plus d'informations, consultez Création d'un collecteur de données.
5. Dans la fenêtre Choisir un collecteur de données, sélectionnez un collecteur de données dans la liste, puis sélectionnez Choisir.
6. La boîte de dialogue Configurer les informations d'identification pour le transfert de données, choisissez Enregistrer.

Sur la page d'accueil du collecteur DMS, dans la carte Transfert de données, vérifiez que les statuts Accès à Amazon S3 et Accès à AWS DMS sont définis sur Oui.

Si vous constatez que le statut Accès à Amazon S3 ou Accès à AWS DMS est défini sur Non, assurez-vous d'avoir créé des ressources IAM pour accéder à Amazon S3 et à DMS Fleet Advisor. Après avoir créé ces ressources IAM avec toutes les autorisations requises, configurez à nouveau le transfert de données. Pour plus d'informations, consultez Création de ressources IAM.

Découverte des serveurs de base de données et de système d'exploitation à surveiller

Vous pouvez utiliser le collecteur de données DMS pour rechercher et répertorier tous les serveurs disponibles dans votre réseau. Il est recommandé de découvrir tous les serveurs de base de données disponibles dans votre réseau, mais ce n'est pas obligatoire. En option, vous pouvez ajouter ou charger manuellement la liste des serveurs pour une collecte de données ultérieure. Pour plus d'informations sur l'ajout manuel d'une liste de serveurs, consultez Gestion des objets surveillés.

Nous vous recommandons de découvrir tous les serveurs de système d'exploitation (OS) avant de découvrir les bases de données sur ces serveurs. Pour découvrir les serveurs du système d'exploitation, vous devez être autorisé à exécuter des scripts et des commandes à distance PowerShell, Secure Shell (SSH) et Windows Management Instrumentation (WMI), ainsi que l'accès au registre Windows. Pour découvrir les serveurs de base de données de votre réseau et collecter des métadonnées à partir de ceux-ci, vous devez disposer d'autorisations d'administrateur en lecture seule pour une connexion à une base de données distante. Assurez-vous d'avoir ajouté un serveur LDAP avant de procéder à la découverte. Pour plus d'informations, consultez Configuration des informations d'identification pour le transfert de données.

Pour commencer à utiliser le collecteur de données DMS, effectuez les tâches suivantes :

- Découvrez tous les serveurs OS de votre réseau.
- Ajoutez des serveurs OS spécifiques en tant qu'objets à surveiller.
- Vérifiez les connexions pour les serveurs de systèmes d'exploitation surveillés.
- Découvrez les bases de données Microsoft SQL Server, MySQL, Oracle et PostgreSQL s'exécutant sur les serveurs OS.
- Ajoutez des serveurs de base de données pour la collecte de données.
- Vérifiez les connexions aux bases de données surveillées.

Pour découvrir les serveurs OS de votre réseau que vous pouvez surveiller

1. Dans le volet de navigation du collecteur de données DMS, choisissez Découverte. Pour afficher le volet de navigation, choisissez l'icône de menu dans l'angle supérieur gauche de la page d'accueil du collecteur de données DMS.

   La page Découverte s'ouvre.

2. Assurez-vous que l'onglet Serveurs de système d'exploitation soit sélectionné, puis choisissez Exécuter la découverte. La boîte de dialogue Paramètres de découverte s'affiche.
3. Entrez les serveurs LDAP que vous souhaitez utiliser pour analyser votre réseau.
4. Choisissez Exécuter la découverte. La page affiche la liste de tous les serveurs OS découverts dans votre réseau, qu'ils exécutent ou non une base de données.

   Nous vous recommandons d'exécuter la découverte pour tous les serveurs OS avant d'exécuter la découverte des bases de données sur ces serveurs. Vos informations d'identification permettent la découverte d'abord pour les serveurs hôtes, puis pour les bases de données qui y résident. Vous devez d'abord découvrir les serveurs OS avant d'exécuter la découverte des bases de données sur ces serveurs. Sachez que les informations d'identification que vous utilisez pour qu'un serveur LDAP trouve des serveurs OS dans votre réseau peuvent être différentes de celles requises pour découvrir des bases de données sur un serveur OS particulier. Par conséquent, nous vous recommandons d'ajouter des serveurs OS aux objets surveillés, de vérifier les informations d'identification et de les corriger si nécessaire, puis de vérifier la connectivité avant de continuer.

Dans la liste des serveurs OS découverts dans votre réseau, vous pouvez désormais sélectionner les serveurs que vous souhaitez ajouter aux objets surveillés.

Pour sélectionner des serveurs OS en tant qu'objets à surveiller

1. Sur la page Découverte, choisissez l'onglet Serveurs de système d'exploitation.
2. Dans la liste des serveurs OS découverts, cochez la case en regard de chaque serveur que vous voulez surveiller.
3. Choisissez Ajouter aux objets surveillés.

Vous pouvez consulter la liste des serveurs OS à surveiller et vérifier les connexions sur la page Surveiller les objets.

Pour vérifier les connexions des serveurs OS sélectionnés à surveiller

1. Dans le volet de navigation du collecteur de données DMS, choisissez Objets surveillés.
2. Sur la page Objets surveillés, choisissez l'onglet Serveurs de système d'exploitation. La liste des serveurs OS découverts à surveiller apparaît.
3. Cochez la case en haut de la colonne pour choisir tous les serveurs OS répertoriés.

4. Choisissez Actions, puis Vérifier la connexion. Pour chaque objet serveur, consultez les résultats dans la colonne Statut des connexions.
5. Sélectionnez les serveurs dont le statut de connexion est autre que Réussite. Ensuite, choisissez Actions, puis Modifier. La boîte de dialogue Modifier le serveur s'ouvre.
6. Vérifiez que les informations sont correctes ou modifiez-les si nécessaire. Lorsque vous avez terminé, choisissez Save (Sauvegarder). La boîte de dialogue Remplacer les informations d'identification s'ouvre.
7. Choisissez Remplacer. Le collecteur de données DMS vérifie et met à jour le statut de chaque connexion sur Réussite.

Vous pouvez désormais découvrir les bases de données qui résident sur les serveurs que vous avez choisi de surveiller.

Pour découvrir les bases de données en cours d'exécution sur les serveurs

1. Dans le volet de navigation du collecteur de données DMS, choisissez Découverte.
2. Choisissez l'onglet Serveurs de base de données, puis Exécuter la découverte. La boîte de dialogue Paramètres de découverte s'affiche.
3. Dans la boîte de dialogue Paramètres de découverte, pour Découverte by, choisissez Objets surveillés. Pour Serveurs, choisissez les serveurs OS sur lesquels vous souhaitez exécuter la découverte de base de données.
4. Choisissez Exécuter la découverte. La page affiche la liste de toutes les bases de données qui résident sur les serveurs OS que vous choisissez de surveiller.

Consultez des informations telles que l'adresse de la base de données, le nom du serveur et le moteur de base de données, qui vous aideront à sélectionner les bases de données à surveiller.

Pour sélectionner les bases de données à surveiller

1. Sur la page Découverte, choisissez l'onglet Serveurs de base de données.
2. Dans la liste des bases de données découvertes, cochez la case en regard de toutes les bases de données que vous souhaitez surveiller.
3. Choisissez Ajouter aux objets surveillés.

Vous pouvez désormais vérifier les connexions aux bases de données que vous choisissez de surveiller.

Pour vérifier les connexions aux bases de données surveillées

1. Dans le volet de navigation du collecteur de données DMS, choisissez Objets surveillés.
2. Sur la page Objets surveillés, choisissez l'onglet Serveurs de base de données. La liste des serveurs de base de données découverts que vous décidez de surveiller apparaît.
3. Cochez la case en haut de la colonne pour choisir tous les serveurs de base de données répertoriés.
4. Choisissez Actions, puis Vérifier la connexion. Pour chaque base de données, consultez les résultats dans la colonne Statut des connexions.
5. Sélectionnez les connexions dont le statut est non défini (vide) ou égal à Échec. Ensuite, choisissez Actions, puis Modifier. La boîte de dialogue Modifier les objets surveillés s'ouvre.
6. Entrez vos informations d'identification Connexion et Mot de passe, puis choisissez Enregistrer. La boîte de dialogue Modifier les informations d'identification s'ouvre.
7. Choisissez Remplacer. Le collecteur de données DMS vérifie et met à jour le statut de chaque connexion sur Réussite.

Après avoir découvert les serveurs OS et les bases de données à surveiller, vous pouvez également effectuer des actions pour gérer les objets surveillés.

Gestion des objets surveillés

Vous pouvez sélectionner les objets à surveiller lorsque vous exécutez le processus de découverte de serveur, comme décrit dans Découverte des serveurs de base de données et de système d'exploitation. Vous pouvez également gérer manuellement les objets, tels que les serveurs de système d'exploitation (OS) et les serveurs de base de données. Vous pouvez effectuer les actions suivantes pour gérer les objets surveillés :

- Ajouter de nouveaux objets à surveiller
- Supprimer des objets existants
- Modifier des objets existants
- Exporter et importer une liste d'objets à surveiller
- Vérifier les connexions aux objets
- Démarrer la collecte des données

Par exemple, vous pouvez ajouter manuellement un objet à surveiller.

Pour ajouter manuellement un objet à surveiller

1. Sur la page Objets surveillés, choisissez +Serveur. La boîte de dialogue Ajouter un objet surveillé s'ouvre.
2. Ajoutez des informations sur le serveur, puis choisissez Enregistrer.

Vous pouvez également utiliser un fichier `.csv` pour importer une grande liste d'objets à surveiller. Utilisez le format de fichier `.csv` suivant pour importer une liste d'objets dans le collecteur de données DMS.

```
Hostname - Hostname or IP address of Monitored Object
Port - TCP port of Monitored Object
Engine: (one of the following)
        • Microsoft SQL Server
        • Microsoft Windows
        • Oracle Database
        • Linux
        • MySQL Server
        • PostgreSQL
Connection type: (one of the following)
        • Login/Password Authentication
        • Windows Authentication
        • Key-Based Authentication
Domain name:(Windows authentication)
        • Use domain name for the account
User name
Password
```

Pour importer un fichier .csv contenant une liste d'objets à surveiller

1. Choisissez Import (Importer). La page Importer des objets surveillés s'ouvre.
2. Naviguez jusqu'au fichier `.csv` que vous voulez importer, puis choisissez Suivant.

Vous pouvez afficher tous les objets et sélectionner ceux à partir desquels vous souhaitez commencer à collecter des métadonnées.

Association d'un serveur OS à une base de données ajoutée manuellement

DMS Fleet Advisor ne peut pas collecter de métriques de performance directement à partir des bases de données MySQL et PostgreSQL. Pour collecter les métriques nécessaires aux recommandations cibles, DMS Fleet Advisor utilise les métriques du système d'exploitation dans lequel vos bases de données s'exécutent.

Lorsque vous ajoutez manuellement des bases de données MySQL et PostgreSQL à la liste des objets surveillés, le collecteur de données DMS ne peut pas identifier les serveurs OS sur lesquels ces bases de données s'exécutent. En raison de ce problème, vous devez associer vos bases de données MySQL et PostgreSQL aux serveurs OS.

Il n'est pas nécessaire d'associer manuellement les serveurs OS aux bases de données que DMS Fleet Advisor a automatiquement découvertes.

Pour associer un serveur OS à la base de données

1. Dans le volet de navigation du collecteur de données DMS, choisissez Objets surveillés.
2. Sur la page Objets surveillés, choisissez l'onglet Serveurs de base de données. La liste des serveurs de base de données s'affiche.
3. Sélectionnez la case à cocher en regard du serveur de base de données MySQL ou PostgreSQL que vous avez ajouté manuellement.
4. Choisissez Actions, puis Modifier. La boîte de dialogue Modifier une base de données s'ouvre.
5. Si votre collecteur de données DMS a déjà découvert le serveur OS sur lequel cette base de données s'exécute, choisissez Détection automatique. Le collecteur de données DMS exécute un script SQL pour identifier automatiquement le serveur OS sur lequel s'exécute la base de données. Le collecteur de données DMS associe ensuite ce serveur OS à la base de données. Ignorez l'étape suivante et enregistrez la configuration de base de données que vous avez modifiée.

   Si le collecteur de données DMS ne parvient pas à identifier automatiquement le serveur OS de la base de données, assurez-vous d'utiliser les informations d'identification correctes et de fournir les autorisations d'accès à la base de données. Éventuellement, vous pouvez ajouter manuellement le serveur OS.
6. Pour ajouter votre serveur OS manuellement, choisissez +Ajouter un serveur OS. La boîte de dialogue Ajouter un serveur OS hôte s'ouvre.

   Ajoutez des informations sur votre serveur OS, puis choisissez Enregistrer.

7. Dans la boîte de dialogue Modifier une base de données, choisissez Vérifier la connexion pour vous assurer que votre collecteur de données DMS peut se connecter au serveur OS.
8. Après avoir vérifié la connexion, choisissez Enregistrer.

Si vous modifiez le serveur OS associé à la base de données source, DMS Fleet Advisor utilise les métriques mises à jour pour générer des recommandations. Cependant, les CloudWatch graphiques Amazon affichent les anciennes données de votre serveur de base de données. Pour plus d'informations sur CloudWatch les graphiques, consultezDétails des recommandations.

Utilisation du protocole SSL avec AWS DMS Fleet Advisor

Pour protéger vos données, AWS DMS Fleet Advisor peut utiliser SSL pour accéder à vos bases de données.

Bases de données prises en charge

AWS DMS Fleet Advisor prend en charge l'utilisation du protocole SSL pour accéder aux bases de données suivantes :

- Microsoft SQL Server
- MySQL
- PostgreSQL

Configuration de SSL

Pour utiliser SSL afin d'accéder à la base de données, configurez votre serveur de base de données pour qu'il prenne en charge SSL. Pour plus d'informations, consultez la documentation suivante pour la base de données :

- SQL Server : Activer les connexions chiffrées au moteur de base de données (langue française non garantie)
- MySQL : Configuration de MySQL pour utiliser des connexions chiffrées (langue française non garantie)
- PostgreSQL : Connexions TCP/IP sécurisées avec SSL

Pour utiliser SSL pour vous connecter à la base de données, sélectionnez Faire confiance au certificat du serveur et Utiliser SSL lors de l'ajout manuel d'un serveur. Pour une base de données

MySQL, vous pouvez utiliser un certificat personnalisé. Pour utiliser un certificat personnalisé, cochez la case Vérifier l'autorité de certification. Pour en savoir plus sur l'ajout d'un serveur, consultez Gestion des objets surveillés.

Vérification du certificat de l'autorité de certification du serveur (CA) pour SQL Server

Si vous souhaitez valider votre certificat d'autorité de certification (CA) de serveur pour SQL Server, désactivez l'option Faire confiance au certificat du serveur quand vous ajoutez le serveur. Si votre serveur utilise une autorité de certification connue et que celle-ci est installée par défaut sur votre système d'exploitation, la vérification devrait fonctionner normalement. Si DMS Fleet Advisor ne parvient pas à se connecter à votre serveur de base de données, installez le certificat CA utilisé par votre serveur de base de données. Pour plus d'informations, consultez Configurer le client.

Collecte de données pour AWS DMS Fleet Advisor

Pour commencer la collecte de données, sélectionnez les objets sur la page Objets surveillés, puis choisissez Exécuter la collecte de données. Le collecteur de données DMS peut collecter jusqu'à 100 bases de données à la fois. En outre, le collecteur de données DMS peut utiliser jusqu'à huit threads parallèles pour se connecter aux bases de données de votre environnement. À partir de ces huit threads, le collecteur de données DMS peut utiliser jusqu'à cinq threads parallèles pour se connecter à une seule instance de base de données.

> **Important**
>
> Avant de commencer à collecter des données, consultez la section Vérification logicielle sur la page d'accueil du collecteur de données DMS. Vérifiez que tous les moteurs de base de données que vous souhaitez surveiller ont le statut Réussite. Si certains moteurs de base de données ont le statut Échec et que vous avez des serveurs de base de données avec des moteurs correspondants dans votre liste d'objets surveillés, corrigez le problème avant de continuer. Vous trouverez des conseils à côté du statut Échec répertorié dans la section Vérification logicielle.

Le collecteur de données DMS peut fonctionner selon deux modes : exécution unique ou surveillance continue. Après avoir démarré la collecte de données, la boîte de dialogue Exécuter la collecte de données s'ouvre. Ensuite, choisissez l'une des deux options suivantes.

Métadonnées et capacité de base de données

Le collecteur de données DMS collecte des informations à partir des serveurs OS ou de base de données. Il inclut les schémas, les versions, les éditions, le CPU, la mémoire et la capacité du disque. Le collecteur de données DMS collecte et fournit également des mesures telles que les IOPS, le débit d'E/S et les connexions actives au serveur de base de données. Vous pouvez calculer les recommandations cibles dans DMS Fleet Advisor sur la base de ces informations. Si la base de données source est surprovisionnée ou sous-provisionnée, les recommandations cibles seront également surprovisionnées ou sous-provisionnées.

Il s’agit de l’option par défaut.

Métadonnées, capacité de base de données et utilisation des ressources

Outre les informations sur les métadonnées et la capacité des bases de données, le collecteur de données DMS collecte des métriques d'utilisation réelles de CPU, de mémoire et de capacité de disque pour les serveurs OS ou de base de données. Le collecteur de données DMS collecte et fournit également des mesures telles que les IOPS, le débit d'E/S et les connexions actives au serveur de base de données. Les recommandations cibles fournies seront plus précises car elles sont basées sur les charges de travail réelles de base de données.

Si vous choisissez cette option, vous définissez la période de collecte des données. Vous pouvez collecter des données au cours des 7 prochains jours ou définir la plage personnalisée de 1 à 60 jours.

Une fois la collecte de données lancée, vous êtes redirigé vers la page de collecte de données, où vous pouvez voir comment les requêtes de collecte s'exécutent et surveillent la progression en temps réel. Vous pouvez consulter l'état général de la collecte sur la page d'accueil du collecteur de données DMS. Si l'état général de la collecte de données est inférieur à 100 %, vous devrez peut-être résoudre les problèmes liés à la collecte.

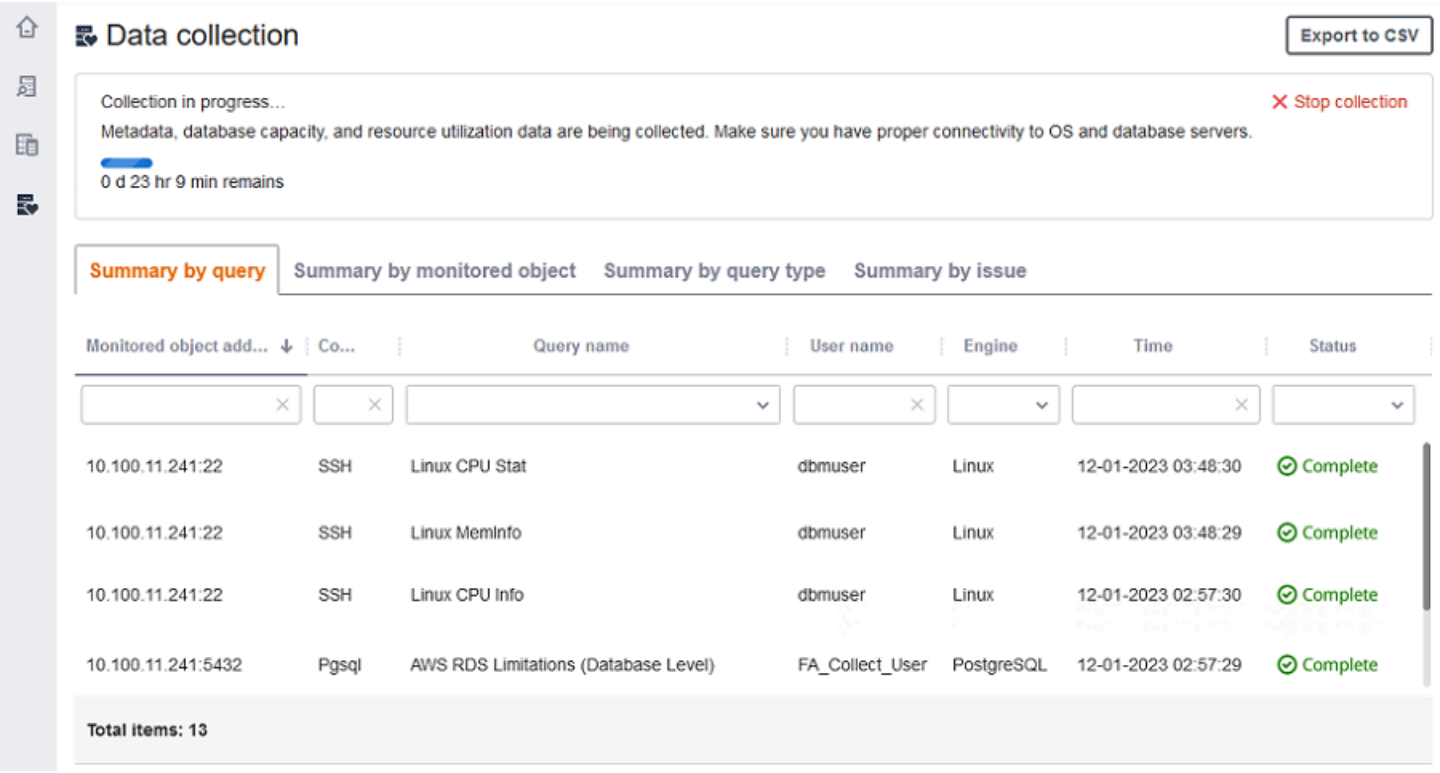
Si vous exécutez le collecteur de données DMS en mode Métadonnées et capacité de base de données, vous pouvez voir le nombre de requêtes terminées sur la page Collecte de données.

Si vous exécutez le collecteur de données DMS en mode Métadonnées, capacité de base de données et utilisation des ressources, vous pouvez voir le temps restant avant que votre collecteur de données DMS termine la surveillance.

Sur la page Collecte de données, vous pouvez voir le statut de la collecte pour chaque objet. Si quelque chose ne fonctionne pas correctement, un message s'affiche indiquant le nombre de

problèmes survenus. Pour vous aider à déterminer une correction pour un problème, vous pouvez vérifier les détails. Les onglets suivants répertorient les problèmes potentiels :

- Résumé par requête : affiche le statut des tests tels que le test Ping. Vous pouvez filtrer les résultats dans la colonne Statut. La colonne Statut fournit un message indiquant le nombre de défaillances survenues lors de la collecte des données.
- Résumé par objet surveillé : montre le statut général par objet.
- Résumé par type de requête : montre le statut pour le type de requête de collecteur, tel que les appels SQL, Secure Shell (SSH) ou Windows Management Instrumentation (WMI).
- Résumé par problème : montre tous les problèmes uniques survenus, avec les noms des problèmes et le nombre de fois où chaque problème s'est produit.



Pour exporter les résultats de collecte, choisissez Exporter au format CSV.

Après avoir identifié les problèmes et les avoir résolus, choisissez Démarrer la collecte et relancez le processus de collecte de données. Après avoir effectué la collecte des données, le collecteur de données utilise des connexions sécurisées pour charger les données collectées dans un inventaire DMS Fleet Advisor. DMS Fleet Advisor stocke les informations dans votre compartiment Amazon S3.

Pour en savoir plus sur la configuration des informations d'identification pour le transfert de données, consultez Configuration des informations d'identification pour le transfert de données.

Collecte de métriques de capacité et d'utilisation des ressources avec AWS DMS Fleet Advisor

Vous pouvez collecter des métadonnées et des métriques de performance selon deux modes : l'exécution unique ou la surveillance continue. Selon l'option que vous sélectionnez, votre collecteur de données DMS suit des métriques différentes dans votre environnement de données. Au cours d'une seule exécution, votre collecteur de données DMS suit uniquement les métriques de métadonnées provenant de vos serveurs OS et de base de données. Au cours de la surveillance continue, votre collecteur de données DMS suit l'utilisation réelle de vos ressources.

AWS DMScollecte les métadonnées et mesures suivantes au cours d'une seule exécution de votre collecteur de données DMS.

- Mémoire disponible sur vos serveurs OS
- Stockage disponible sur vos serveurs OS
- Version et édition de base de données
- Nombre de CPU sur vos serveurs OS
- Nombre de schémas
- Nombre de procédures stockées
- Nombre de tables
- Nombre de déclencheurs
- Nombre de vues
- Structure des schémas

DMS Fleet Advisor utilise ces métriques pour établir un inventaire de vos serveurs OS et de base de données. DMS Fleet Advisor utilise également ces métadonnées et mesures pour analyser les schémas de votre base de données source.

DMS Fleet Advisor peut générer des recommandations cibles à l'aide des métriques collectées lors d'une seule exécution du collecteur de données. Toutefois, dans ce cas, pour vos bases de données sources surapprovisionnées, la recommandation cible est également surprovisionnée. Ainsi, vous encourez des coûts supplémentaires pour la maintenance de vos ressources dans leAWS Cloud.

Pour les bases de données sources sous-provisionnées, la recommandation cible est également sous-provisionnée, ce qui peut entraîner des problèmes de performances. Nous recommandons de collecter les données à l'aide d'une surveillance continue en choisissant les métadonnées, la capacité de la base de données et le mode d'utilisation des ressources pour le collecteur de données DMS.

AWS DMS rassemble les métriques suivantes dans le cadre de la surveillance continue. Vous pouvez exécuter votre collecteur de données DMS pendant une période de 1 à 60 jours.

- Débit d'E/S sur vos serveurs de base de données
- Opérations d'entrée/sortie par seconde (IOPS) sur vos serveurs de base de données
- Nombre de CPU utilisés par vos serveurs OS
- Utilisation de la mémoire sur vos serveurs OS
- Nombre de connexions actives à la base de données et au serveur du système d'exploitation

DMS Fleet Advisor utilise ces métriques pour générer des recommandations cibles précises, afin que vos bases de données cibles répondent à vos besoins de performances. Cela peut éviter des coûts supplémentaires liés à la maintenance de vos ressources dans leAWS Cloud.

Comment AWS DMS Fleet Advisor collecte-t-il les métriques de capacité et d'utilisation des ressources ?

DMS Fleet Advisor collecte des métriques de performance chaque minute.

Pour Oracle et SQL Server, DMS Fleet Advisor exécute des requêtes SQL pour capturer la valeur de chaque métrique de base de données.

Pour MySQL et PostgreSQL, DMS Fleet Advisor collecte des métriques de performance à partir du serveur OS sur lequel s'exécute la base de données. Dans Windows, DMS Fleet Advisor exécute des scripts WQL (WMI Query Language) et reçoit des données WMI. Dans Linux, DMS Fleet Advisor exécute des commandes qui capturent les métriques des serveurs OS.

> **Important**
>
> L'exécution de scripts SQL distants peut avoir un impact sur les performances de vos bases de données de production. Toutefois, les requêtes de collecte de données ne contiennent aucune logique de calcul. Il est donc peu probable que le processus de collecte de données utilise plus de 1 % de vos ressources de base de données.

Vous pouvez consulter toutes les requêtes que le collecteur de données exécute pour collecter les métriques. Pour ce faire, ouvrez le fichier `DMSCollector.Collections.json`. Ce fichier se trouve dans le dossier `etc`, qui est situé dans le dossier où vous avez installé le collecteur de données. Le chemin par défaut est `C:\ProgramData\Amazon\AWS DMS Collector\etc \DMSCollector.Collections.json`.

Le collecteur de données DMS utilise le système de fichiers local comme stockage temporaire pour toutes les données collectées. Le collecteur de données DMS stocke les données collectées au format JSON. Vous pouvez utiliser le collecteur local en mode hors connexion et vérifier manuellement les fichiers collectés avant de configurer le transfert de données. Vous pouvez voir tous les fichiers collectés dans le dossier `out` situé dans le dossier où vous avez installé le collecteur de données DMS. Le chemin par défaut est `C:\ProgramData\Amazon\AWS DMS Collector \out`.

> **Important**
>
> Si vous utilisez votre collecteur de données DMS en mode hors ligne et que vous stockez les données collectées sur votre serveur pendant plus de 14 jours, vous ne pouvez pas utiliser Amazon CloudWatch pour afficher ces statistiques. Toutefois, DMS Fleet Advisor utilise toujours ces données pour générer des recommandations. Pour plus d'informations sur CloudWatch les graphiques, consultezDétails des recommandations.

Vous pouvez également vérifier les fichiers de données collectés dans un mode en ligne. Le collecteur de données DMS transfère toutes les données vers le compartiment Amazon S3 que vous avez spécifié dans les paramètres du collecteur de données DMS.

Vous pouvez utiliser votre collecteur de données DMS pour collecter des données à partir de bases de données sur site. Vous pouvez également collecter des données à partir de bases de données Amazon RDS et Aurora. Toutefois, vous ne pouvez pas exécuter correctement toutes les requêtes du collecteur de données DMS dans le cloud en raison des différences entre Amazon RDS ou Aurora et les instances de base de données sur site. Étant donné que le collecteur de données DMS rassemble les métriques d'utilisation des bases de données MySQL et PostgreSQL à partir du système d'exploitation hôte, cette approche ne fonctionnera pas avec Amazon RDS et Aurora.

Résolution des problèmes liés au collecteur de données DMS

Dans la liste suivante, vous pouvez rechercher les actions à entreprendre lorsque vous rencontrez des problèmes spécifiques lorsque vous collectez des données avec votre collecteur de données.

Rubriques

- Problèmes de collecte de données liés aux connexions réseau et serveur
- Problèmes de collecte de données liés à Windows Management Instrumentation
- Problèmes de collecte de données liés au compositeur de pages Web Windows
- Problèmes de collecte de données liés à SSL

Problèmes de collecte de données liés aux connexions réseau et serveur

NET : une exception s'est produite lors d'une demande ping.

Vérifiez le nom de l'ordinateur pour voir s'il est dans un état où il ne peut pas être résolu en adresse IP.

Par exemple, vérifiez si l'ordinateur est éteint, déconnecté du réseau ou mis hors service.

NET : dépassement de délai

Activez la règle de pare-feu entrant « Partage de fichiers et d'imprimantes (Demande d'écho - ICMPv4-In) ». Par exemple :

```
* Inbound ICMPv4
```

FILET : DestinationHostUnreachable

Vérifiez l'adresse IP de l'ordinateur. Plus précisément, vérifiez s'il se trouve dans le même sous-réseau que l'ordinateur qui exécute le collecteur de données DMS et s'il répond aux demandes ARP (Address Resolution Protocol).

Si l'ordinateur se trouve dans un autre sous-réseau, l'adresse IP de la passerelle ne peut pas être résolue avec l'adresse MAC (Media Access Control).

De plus, vérifiez si l'ordinateur est éteint, déconnecté du réseau ou mis hors service.

Problèmes de collecte de données liés à Windows Management Instrumentation

WMI : le serveur RPC n'est pas disponible. (Exception de HRESULT : 0x800706BA)

Activez la règle de pare-feu entrant « Windows Management Instrumentation (DCOM–In) ». Par exemple :

```
* Inbound TCP/IP at local port 135.
```

Activez également la règle de pare-feu entrant « Windows Management Instrumentation (WMI–In) ». Par exemple :

`* Inbound TCP/IP at local port 49152 – 65535` pour Windows Server 2008 et versions ultérieures.

`* Inbound TCP/IP at local port 1025 – 5000` pour Windows Server 2003 et versions antérieures.

WMI : accès refusé. (Exception de HRESULT : 0x80070005 (E_ACCESSDENIED))

Essayez les éléments suivants :

- Ajoutez l'utilisateur du collecteur de données DMS au groupe Windows, aux utilisateurs COM distribués ou aux administrateurs.
- Démarrez le service Windows Management Instrumentation et définissez son type de démarrage sur Automatique.
- Assurez-vous que le nom d'utilisateur de votre collecteur de données DMS est au format \.

WMI : accès refusé

Ajoutez l'autorisation Appel à distance autorisé à l'utilisateur du collecteur de données DMS sur l'espace de noms WMI racine.

Utilisez les paramètres avancés et assurez-vous que les autorisations s'appliquent à « Cet espace de noms et les sous-espaces ».

WMI : l'appel a été annulé par le filtre de messages. (Exception de HRESULT : 0x80010002...)

Redémarrez le service Windows Management Instrumentation.

Problèmes de collecte de données liés au compositeur de pages Web Windows

WPC : le chemin réseau est introuvable

Activez la règle de pare-feu entrant « Partage de fichiers et d'imprimantes (SMB–In) ». Par exemple :

`* Inbound TCP/IP at local port 445`.

Démarrez également le service Registre à distance et définissez son type de démarrage sur Automatique.

WPC : accès refusé

Ajoutez l'utilisateur du collecteur de données DMS au groupe Utilisateurs ou administrateurs de l'Analyseur de performances.

WPC : la catégorie n'existe pas

Exécutez `loader /r` pour reconstruire le cache du compteur de performances, puis redémarrez votre ordinateur.

> **Note**
>
> Pour en savoir plus sur la résolution des problèmes liés à la migration de données à l'aide d'AWS Database Migration Service (AWS DMS), consultez Résolution des problèmes et assistance au diagnostic.

Problèmes de collecte de données liés à SSL

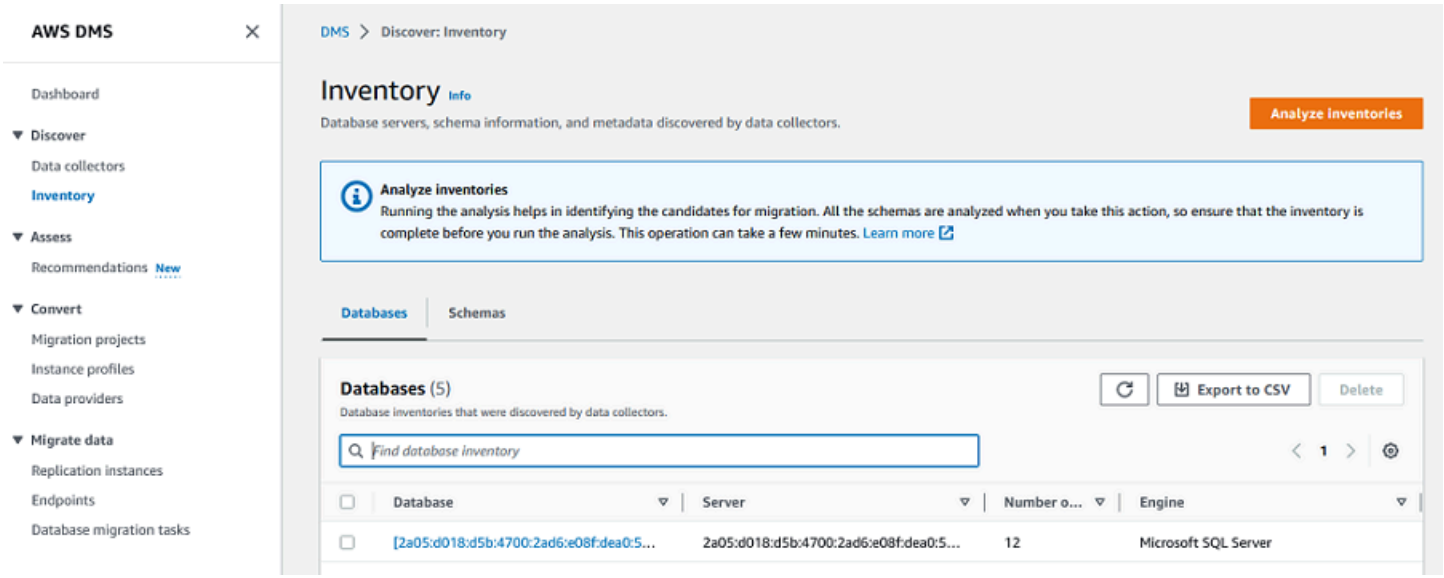
Erreurs SSL

Votre base de données nécessite une connexion SSL sécurisée et vous n'avez pas activé les options Vérifier l'autorité de certification et Utiliser SSL pour la connexion. Activez ces options et assurez-vous que l'autorité de certification utilisée par la base de données est installée sur votre système d'exploitation local. Pour plus d'informations, consultez Configuration de SSL.

Utilisation des inventaires à des fins d'analyse dans AWS DMS Fleet Advisor

Pour vérifier la faisabilité d'éventuelles migrations de bases de données, vous pouvez utiliser les inventaires des bases de données et schémas découverts. Vous pouvez utiliser les informations contenues dans ces inventaires pour identifier les bases de données et les schémas qui sont de bons candidats à la migration.

Vous pouvez accéder aux inventaires des bases de données et des schémas dans la console. Pour ce faire, choisissez Inventaire dans la console.



DMS Fleet Advisor analyse vos schémas de base de données pour déterminer la similarité des différents schémas. Cette analyse ne compare pas le code réel des objets. DMS Fleet Advisor compare uniquement les noms des objets de schéma, tels que les fonctions et les procédures, afin d'identifier des objets similaires dans différents schémas de base de données.

Rubriques

- Utilisation d'un inventaire des bases de données à des fins d'analyse
- Utilisation d'un inventaire des schémas à des fins d'analyse

Utilisation d'un inventaire des bases de données à des fins d'analyse

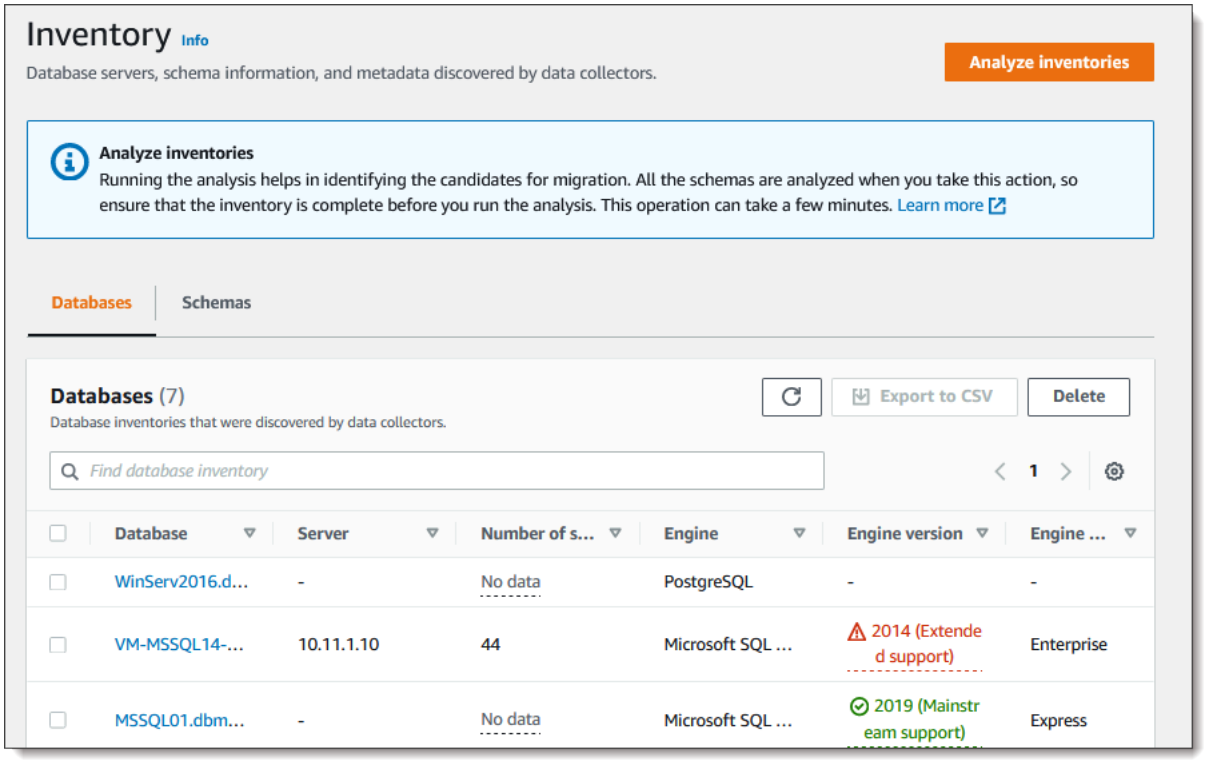
Pour consulter la liste de toutes les bases de données présentes sur tous les serveurs découverts de votre réseau à partir desquels des données ont été collectées, procédez comme suit.

Pour afficher la liste des bases de données figurant sur les serveurs de votre réseau à partir desquels des données ont été collectées

1. Choisissez Inventaire dans la console.

   La page Inventaire s'ouvre.
2. Sélectionnez l'onglet Databases (Bases de données).

   La liste des bases de données découvertes s'affiche.



3. Choisissez Analyser les inventaires pour déterminer les propriétés des schémas, telles que la similarité et la complexité. La durée du processus dépend du nombre d'objets à analyser, mais cela ne prendra pas plus d'une heure. Les résultats de l'analyse se trouvent dans l'onglet Schémas de la page Inventaire.

   DMS Fleet Advisor analyse les schémas de toutes les bases de données découvertes afin de définir l'intersection de leurs objets. Le résultat de l'analyse est exprimé en pourcentage. DMS Fleet Advisor considère les schémas présentant des intersections de plus de 50 % comme des doublons. Le schéma d'origine est identifié comme le schéma dans lequel des doublons ont été trouvés. Cela permet d'identifier les schémas originaux à convertir ou à migrer en premier.

   L'ensemble de l'inventaire est analysé ensemble pour identifier les schémas dupliqués.

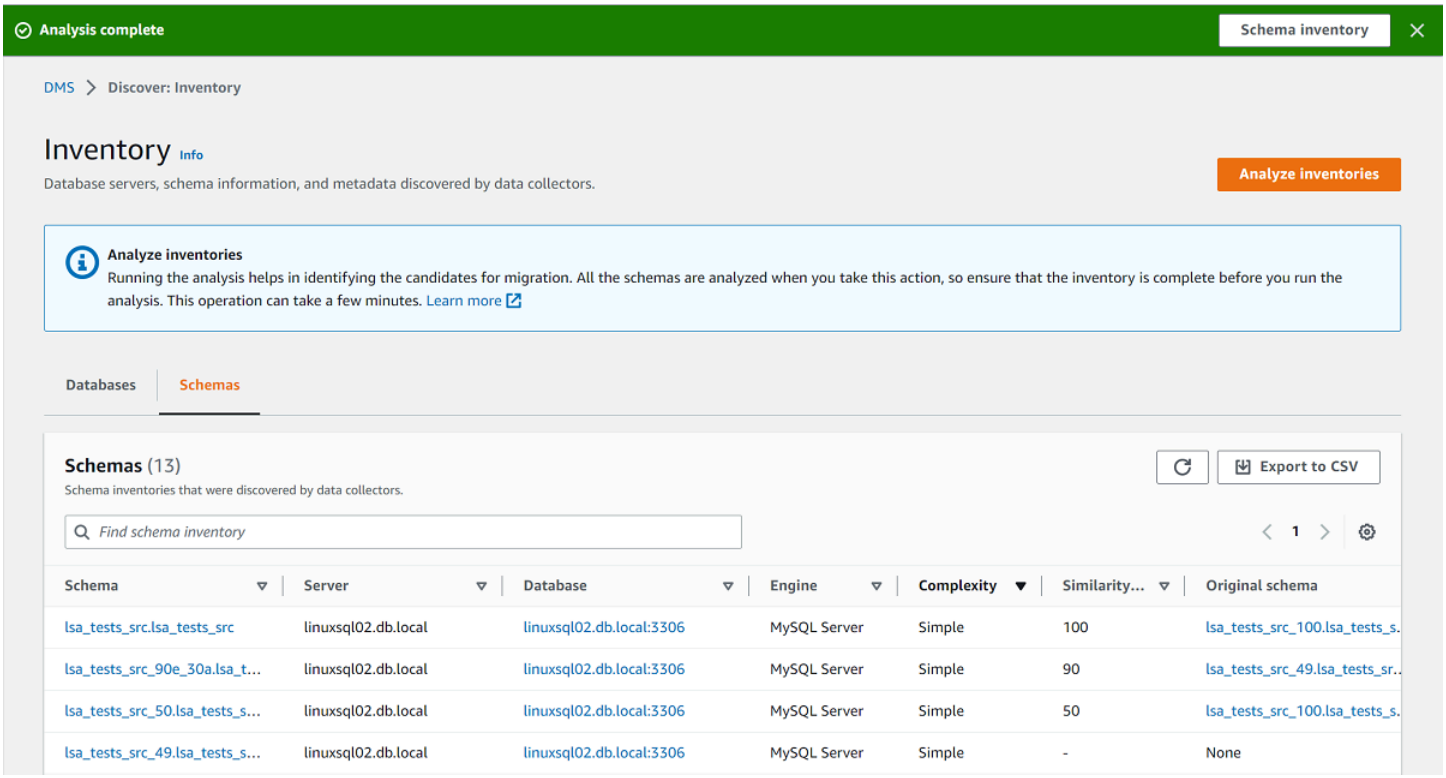
Utilisation d'un inventaire des schémas à des fins d'analyse

Vous pouvez consulter la liste des schémas de base de données découverts sur les serveurs de votre réseau à partir desquels les données ont été collectées. Utilisez la procédure suivante.

Pour afficher la liste des schémas figurant sur les serveurs de votre réseau à partir desquels des données ont été collectées

1. Choisissez Inventaire dans la console. La page Inventaire s'ouvre.
2. Choisissez l'onglet Schémas. La liste des schémas s'affiche.
3. Sélectionnez un schéma dans la liste pour afficher les informations le concernant, notamment le serveur, la base de données, la taille et la complexité.

   Pour chaque schéma, vous pouvez consulter un résumé des objets qui fournit des informations sur les types d'objets, le nombre d'objets, la taille des objets et les lignes de code.
4. (Facultatif) Choisissez Analyser les inventaires pour identifier les schémas dupliqués. DMS Fleet Advisor analyse les schémas de base de données pour définir l'intersection de leurs objets.
5. Vous pouvez exporter les informations d'inventaire vers un fichier `.csv` pour un examen plus approfondi.



Pour identifier les schémas à migrer et déterminer la cible de migration, vous pouvez utiliser AWS Schema Conversion Tool (AWS SCT) ou DMS Schema Conversion. Pour plus d'informations, consultez Utilisation d'un nouvel assistant de projet dans AWS SCT.

Après avoir identifié les schémas à migrer, vous pouvez les convertir à l'aide d'AWS SCT ou de la conversion de schéma DMS. Pour plus d'informations sur la conversion de schéma DMS, consultez Conversion de schémas de base de données à l'aide de la conversion de schéma DMS.

Utilisation de la fonctionnalité de recommandations cibles d'AWS DMS Fleet Advisor

Pour explorer et choisir une cible de migration optimale, vous pouvez générer des recommandations cibles pour vos bases de données sources sur site dans DMS Fleet Advisor. Une recommandation inclut un ou plusieurs moteurs AWS cibles possibles que vous pouvez choisir pour la migration de la base de données source sur site. Parmi ces moteurs cibles possibles, DMS Fleet Advisor suggère un moteur cible unique comme destination de migration de taille appropriée et indique que cette cible est recommandée par DMS. Pour déterminer cette destination de migration de taille adaptée, DMS Fleet Advisor utilise les métadonnées d'inventaire et les métriques collectées par votre collecteur de données.

Vous pouvez utiliser des recommandations avant le début d'une migration pour découvrir les options de migration, réduire les coûts et réduire les risques. Vous pouvez exporter les recommandations sous la forme d'un fichier CSV (de valeurs séparées par des virgules) et le partager avec les principales parties prenantes pour faciliter la prise de décision. Vous pouvez exporter les recommandations vers le AWS Pricing Calculator afin d'optimiser davantage les coûts de maintenance. Pour plus d'informations, consultez https://calculator.aws/#/.

Vous ne pouvez pas modifier les recommandations cibles dans DMS Fleet Advisor. Vous ne pouvez donc pas utiliser DMS Fleet Advisor pour effectuer des analyses hypothétiques. Une analyse hypothétique consiste à modifier les paramètres cibles pour voir comment ces modifications affectent l'estimation du prix de votre recommandation. Vous pouvez exécuter une analyse hypothétique dans le AWS Pricing Calculator en utilisant les paramètres cibles recommandés comme point de départ dans le AWS Pricing Calculator. Pour plus d'informations, consultez https://calculator.aws/#/.

Nous vous recommandons de considérer la recommandation de DMS Fleet Advisor comme point de départ dans la planification de votre migration. Vous pouvez ensuite décider de modifier les paramètres d'instance recommandés afin d'optimiser le coût ou les performances des charges de travail de la base de données.

Rubriques

- Instances cibles recommandées

- Comment DMS Fleet Advisor détermine-t-il les spécifications des instances cibles pour la recommandation ?
- Génération de recommandations cibles avec AWS DMS Fleet Advisor
- Exploration des détails des recommandations cibles avec AWS DMS Fleet Advisor
- Exportation des recommandations cibles avec AWS DMS Fleet Advisor
- Découvrir et analyser les limites de la migration avec AWS DMS Fleet Advisor
- Résolution des problèmes liés aux recommandations cibles

Instances cibles recommandées

Pour les recommandations cibles, DMS Fleet Advisor prend en compte les instances de base de données Amazon RDS à usage général, à mémoire optimisée et à capacité extensible suivantes.

- db.m5
- db.m6i
- db.r5
- db.r6i
- db.t3
- db.x1
- db.x1e
- db.z1d

Pour plus d'informations sur les classes d'instances de base de données Amazon RDS, consultez Classes d'instances de base de données dans le Guide de l'utilisateur Amazon RDS.

Comment DMS Fleet Advisor détermine-t-il les spécifications des instances cibles pour la recommandation ?

DMS Fleet Advisor peut générer des recommandations en fonction de la capacité ou de l'utilisation des bases de données.

- Si vous choisissez de générer la recommandation en fonction de la capacité des bases de données, DMS Fleet Advisor mappe la capacité des bases de données existante aux spécifications de la classe d'instances la plus proche.

- Si vous choisissez de générer la recommandation en fonction de l'utilisation des ressources, DMS Fleet Advisor détermine la valeur du 95e centile pour des métriques telles que le CPU, la mémoire, le débit d'E/S et les IOPS. Le 95e percentile signifie que 95 % des données collectées sont inférieures à cette valeur. Ensuite, DMS Fleet Advisor mappe ces valeurs aux spécifications de la classe d'instances la plus proche.

Pour déterminer la taille de la base de données cible, DMS Fleet Advisor collecte des informations sur la taille de la base de données source. DMS Fleet Advisor recommande ensuite d'utiliser la même taille pour le stockage cible. Si le stockage de la base de données source est surprovisionné, la taille recommandée du stockage cible sera également surprovisionnée.

Si vous souhaitez migrer des données à l'aide d'AWS DMS, vous devrez peut-être augmenter le provisionnement d'IOPS pour votre instance de base de données cible. Lorsque DMS Fleet Advisor génère des recommandations cibles, le service prend en compte uniquement les métriques de la base de données source. DMS Fleet Advisor ne prend pas en compte les IOPS supplémentaires dont vous pourriez avoir besoin pour exécuter des tâches de migration de données. Pour plus d'informations, consultez Les tâches de migration s'exécutent lentement.

Pour estimer les coûts d'IOPS, DMS Fleet Advisor utilise un one-to-one mappage de l'utilisation des IOPS de votre source comme référence. DMS Fleet Advisor considère le pic de charge comme la valeur de référence et le taux d'utilisation de 100 % pour la tarification des IOPS.

Pour les bases de données sources PostgreSQL et MySQL, DMS Fleet Advisor peut inclure les instances de base de données Aurora et Amazon RDS dans les recommandations cibles. Si une configuration Aurora correspond aux exigences de la source, DMS Fleet Advisor marque cette option comme recommandée.

Génération de recommandations cibles avec AWS DMS Fleet Advisor

Une fois que vous avez terminé la collecte des données et l'inventaire de la base de données et de votre analytique de flotte, vous pouvez générer des recommandations cibles dans DMS Fleet Advisor. Pour ce faire, choisissez les bases de données sources et configurez les paramètres utilisés par la fonctionnalité Recommandations cibles de DMS Fleet Advisor pour déterminer la taille des instances cibles. En outre, la fonctionnalité Recommandations cibles de DMS Fleet Advisor utilise les métriques de capacité et d'utilisation collectées à partir de vos bases de données sources.

Pour générer des recommandations cibles

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.

   Assurez-vous de choisir la Région AWS dans laquelle vous utilisez DMS Fleet Advisor.
2. Dans le volet de navigation, choisissez Recommandations sous Évaluer, puis choisissez Générer des recommandations.
3. Dans le volet Sélectionnez les bases de données sources, cochez les cases correspondant aux noms des bases de données que vous souhaitez migrer vers le AWS Cloud.

   Pour Rechercher des bases de données sources, entrez le nom de la base de données pour filtrer votre inventaire.

   DMS Fleet Advisor peut générer des recommandations pour un maximum de 100 bases de données à la fois.
4. Pour Disponibilité et durabilité, choisissez l'option de déploiement préférée.

   Pour calculer les recommandations cibles pour vos bases de données de production, choisissez Production (Multi-AZ). DMS Fleet Advisor inclut deux instances de base de données dans différentes zones de disponibilité dans votre recommandation cible. Cette option de déploiement multi-AZ fournit une haute disponibilité, une redondance des données et une prise en charge du basculement.

   Si Aurora est le moteur cible recommandé et si Availability and Durability est un déploiement multi-AZ, la recommandation cible inclut une instance de base de données de lecture et d'écriture.

   Pour calculer les recommandations cibles pour les bases de données que vous utilisez pour le développement ou les tests, choisissez Dev/Test (Mono-AZ). DMS Fleet Advisor inclut une seule instance de base de données dans votre recommandation cible. Cette option de déploiement mono-AZ réduit les coûts de maintenance.
5. Pour Dimensionnement de l'instance cible, choisissez l'option préférée que DMS Fleet Advisor utilise pour calculer les recommandations cibles.

   Pour calculer les recommandations cibles en fonction de la configuration de votre serveur OS ou base de données source, choisissez Capacité totale. DMS Fleet Advisor utilise des métriques telles que le nombre total de CPU, la mémoire et la capacité de disque de vos serveurs OS ou

de base de données source pour générer des recommandations cibles. Ensuite, DMS Fleet Advisor mappe les métriques de capacité de la base de données sur les spécifications de la classe d'instances de base de données Amazon RDS la plus proche.

Pour calculer les recommandations cibles en fonction de l'utilisation réelle de votre serveur OS ou base de données source, choisissez Utilisation des ressources. DMSFleet Advisor utilise les métriques d'utilisation de CPU, de mémoire et de capacité de disque de vos serveurs OS ou de base de données source pour générer des recommandations cibles. À partir des métriques d'utilisation, DMS Fleet Advisor calcule le 95e centile pour chaque métrique. Le 95e centile signifie que 95 % des données de cette période sont inférieures à cette valeur. Ensuite, DMS Fleet Advisor mappe ces valeurs à la classe d'instances de base de données Amazon RDS la plus proche.

Nous vous recommandons d'utiliser l'option Utilisation des ressources pour des recommandations plus précises. Pour ce faire, assurez-vous d'avoir collecté les métriques de capacité totale et d'utilisation des ressources.

6. Sélectionnez Generate (Générer).

DMS Fleet Advisor génère des recommandations cibles pour les bases de données sélectionnées. Pour les recommandations générées avec succès, DMS Fleet Advisor définit le statut sur Calculé. DMS Fleet Advisor utilise également le AWS Pricing Calculator pour déterminer le coût mensuel estimé de l'instance de base de données cible recommandée. Vous pouvez maintenant explorer en détail les recommandations générées. Pour plus d'informations, consultez Détails des recommandations.

Pour estimer le coût mensuel total de votre inventaire de données, cochez les cases correspondant aux bases de données que vous prévoyez de déplacer vers le cloud. DMS Fleet Advisor affiche le coût mensuel total estimé et le résumé de vos bases de données cibles dans le AWS Cloud. DMS Fleet Advisor utilise l'API AWS Price List Query pour fournir des détails de tarification à titre informatif uniquement. Vos frais réels dépendent de divers facteurs, notamment de votre utilisation réelle des Services AWS. Pour plus d'informations sur la tarification des Service AWS, consultez Tarification des services cloud.

Exploration des détails des recommandations cibles avec AWS DMS Fleet Advisor

Une fois que DMS Fleet Advisor a généré des recommandations cibles, vous pouvez consulter les paramètres clés de la cible de migration recommandée dans le tableau Recommandations. Ces paramètres clés incluent le moteur cible, la classe d'instances, le nombre de CPU virtuels, la mémoire, le stockage et le type de stockage. Outre ces paramètres, DMS Fleet Advisor affiche le coût mensuel estimé de cette cible de migration recommandée.

Chaque recommandation peut inclure un ou plusieurs moteurs cibles AWS possibles. Si votre recommandation inclut plusieurs moteurs cibles, AWS DMS marque l'un d'entre eux comme recommandé. De plus, AWS DMS indique les paramètres et le coût mensuel estimé pour cette option recommandée dans le tableau Recommandations.

Pour comparer les recommandations cibles à l'utilisation et à la capacité de la base de données source, explorez vos recommandations en détail. Vous pouvez également consulter les limitations de migration pour une recommandation sélectionnée. Ces limitations incluent des fonctionnalités de base de données non prises en charge, des éléments d'action et d'autres considérations relatives à la migration.

Pour explorer la recommandation en détail

1. Générez des recommandations cibles avec DMS Fleet Advisor. Pour plus d'informations, consultez Génération de recommandations cibles.
2. Choisissez le nom de la recommandation dans le tableau Recommandations. La page de la recommandation s'ouvre.
3. Si la recommandation inclut plusieurs options cibles, choisissez l'option cible dans Recommandations cibles.
4. Développez la section Utilisation et capacités de la source. DMS Fleet Advisor affiche des graphiques d'utilisation des ressources pour les métriques suivantes.

    - Nombre de processeurs
    - Mémoire
    - Débit d'E/S
    - Nombre d'opérations d'entrée/sortie par seconde (IOPS)
    - Stockage
    - Nombre de connexions actives au serveur de base de données

Utilisez ces graphiques pour comparer les métriques de la base de données source issues de votre collecteur de données DMS aux métriques du moteur cible sélectionné.

Si les graphiques ne s'affichent pas après avoir développé la section Utilisation et capacité de la source, assurez-vous d'avoir autorisé votre utilisateur IAM à consulter les CloudWatch tableaux de bord Amazon. Pour plus d'informations, consultez la section Utilisation CloudWatch des tableaux de bord Amazon dans le guide de CloudWatch l'utilisateur Amazon.

5. Choisissez le lien avec le nom du moteur cible que vous avez sélectionné. La page Détails de la cible s'ouvre.
6. Pour exporter les recommandations cibles au format CSV, choisissez l'option Exporter au format CSV dans le menu déroulant Actions.
7. Pour exporter les recommandations cibles versAWS Pricing Calculator, choisissez l'AWS Pricing Calculatoroption Optimiser les coûts avec dans le menu déroulant Actions.
8. Dans la section Configuration, comparez les valeurs des paramètres de la base de données source aux paramètres du moteur cible. Pour le moteur cible, DMS Fleet Advisor affiche les coûts mensuels estimés pour vos ressources cloud. DMS Fleet Advisor utilise l'API AWS Price List Query pour fournir des détails de tarification à titre informatif uniquement. Vos frais réels dépendent de divers facteurs, notamment de votre utilisation réelle des Services AWS. Pour plus d'informations sur la tarification des Service AWS, consultez la tarification des services cloud (https://aws.amazon.com/pricing/).
9. Dans la section Limites de migration, consultez les limites de migration. Nous vous recommandons de tenir compte de ces limitations lorsque vous migrez la base de données source vers le AWS Cloud.

Exportation des recommandations cibles avec AWS DMS Fleet Advisor

Après avoir généré des recommandations cibles, vous pouvez enregistrer une copie de la liste des recommandations sous forme de fichier CSV (valeurs séparées par des virgules).

Pour générer des recommandations cibles

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.

   Assurez-vous de choisir la Région AWS dans laquelle vous utilisez DMS Fleet Advisor.

2. Dans le volet de navigation, choisissez Recommandations sous Évaluer, puis sélectionnez les recommandations à inclure dans votre fichier CSV.
3. Choisissez Exporter au format CSV, entrez le nom du fichier et choisissez le dossier sur votre PC dans lequel enregistrer ce fichier.
4. Ouvrez le fichier CSV.

Le fichier CSV de recommandations contient les informations suivantes.

- CreatedDate— Date à laquelle DMS Fleet Advisor a créé la recommandation du moteur cible.
- DatabaseId— L'identifiant de la base de données source pour laquelle DMS Fleet Advisor a créé cette recommandation.
- DeploymentOption— L'option de déploiement pour l'instance de base de données Amazon RDS recommandée.
- EngineEdition— L'édition cible du moteur Amazon RDS recommandée.
- EngineName— Nom du moteur cible.
- InstanceMemory— La quantité de mémoire sur l'instance de base de données Amazon RDS recommandée.
- InstanceSizingType— La taille de votre instance cible.
- InstanceType— Le type d'instance Amazon RDS cible recommandé.
- InstanceVcpu— Le nombre de processeurs virtuels sur l'instance de base de données Amazon RDS recommandée.
- Preferred : indicateur booléen indiquant que cette option cible est recommandée.
- Status : statut de la recommandation de moteur cible.
- StorageIops— Le nombre d'opérations d'E/S effectuées chaque seconde (IOPS) sur l'instance de base de données Amazon RDS recommandée.
- StorageSize— La taille de stockage de l'instance de base de données Amazon RDS recommandée.
- StorageType— Le type de stockage de l'instance de base de données Amazon RDS recommandée.
- WorkloadType— L'option de déploiement pour votre moteur cible, telle que le déploiement multi-AZ ou mono-AZ.

Découvrir et analyser les limites de la migration avec AWS DMS Fleet Advisor

Vous pouvez utiliser le collecteur de données DMS pour découvrir les fonctionnalités de base de données que votre moteur cible ne prend pas en charge. Pour choisir la bonne cible de migration, vous devez tenir compte de ces limitations.

Le collecteur de données DMS découvre les fonctionnalités spécifiques de la base de données source. Ensuite, DMS Fleet Advisor analyse les fonctionnalités de la source du point de vue de la migration vers la cible spécifiée et fournit des informations supplémentaires sur la limitation et inclut des actions recommandées pour remédier à cette limitation ou l'éviter. DMS Fleet Advisor calcule également l'impact de ces limitations.

La liste des limitations est disponible sur la page de détails du moteur cible. Accédez à cette page depuis la page Recommandations dans le menu de navigation de gauche. Dans la liste des cibles, choisissez le moteur cible à examiner. La liste des restrictions se trouve au bas de la page.

Le tableau suivant inclut les fonctionnalités de base de données MySQL qu'Amazon RDS for MySQL ne prend pas en charge.

Limitation	Description	Impact
Plug-ins d'authentification	Amazon RDS ne prend pas en charge les plug-ins d'authentification MySQL.	Faible
Journalisation des erreurs dans le journal système	Amazon RDS ne prend pas en charge l'écriture du journal des erreurs dans le journal système.	Faible
Identifiants de transaction globaux	Vous pouvez utiliser des identifiants de transaction globaux avec toutes les versions de RDS for MySQL 5.7 et RDS for MySQL version 8.0.26 et les versions ultérieures de MySQL 8.0.	Faible

Limitation	Description	Impact
Réplication de groupe	Amazon RDS ne prend pas en charge le plug-in MySQL Group Replication.	Faible
Chiffrement d'espace de table InnoDB	Amazon RDS ne prend pas en charge le chiffrement d'espace de table InnoDB.	Faible
Mot réservé InnoDB	InnoDB est un mot réservé pour Amazon RDS for MySQL. Vous ne pouvez pas utiliser ce nom pour une base de données MySQL.	Faible
Plug-in Keyring	Amazon RDS ne prend pas en charge le plug-in MySQL Keyring.	Faible
plugin de validation de mot de passe	Amazon RDS ne prend pas en charge le plug-in MySQL `validate_password` .	Faible
Variables système persistantes	Amazon RDS ne prend pas en charge les variables système persistantes MySQL.	Faible

Limitation	Description	Impact
Accès limité	Amazon RDS restreint également l'accès à certaines procédures système et tables qui requièrent des autorisat ions avancées. De plus, Amazon RDS ne permet pas l'accès hôte direct à une instance de base de données en utilisant Telnet, Secure Shell (SSH) ou une connexion Bureau à distance Windows.	Faible
Plug-in de réécriture de requêtes Rewriter	Amazon RDS ne prend pas en charge le plug-in de réécriture de requêtes MySQL Rewriter.	Faible
Réplication semi-synchrone	Amazon RDS ne prend pas en charge la réplication semi-synchrone MySQL.	Faible
Espaces de table transport ables	Amazon RDS ne prend pas en charge la fonctionnalité MySQL d'espaces de table transportables.	Faible
Plug-in X	Amazon RDS ne prend pas en charge le plug-in MySQL X.	Faible

Le tableau suivant inclut les fonctionnalités de base de données Oracle qu'Amazon RDS for Oracle ne prend pas en charge.

Limitation	Description	Impact
Active Data Guard	Vous ne pouvez pas utiliser Active Data Guard avec	Medium

Limitation	Description	Impact
	les bases de données de conteneurs (CDB) multiloca taires Oracle.	
Automatic Storage Management	Amazon RDS ne prend pas en charge Oracle Automatic Storage Management (Oracle ASM).	Medium
Flux d'activité de base de données	Amazon RDS ne prend pas en charge les flux d'activité de base de données Oracle pour l'architecture à locataire unique.	Élevée
Limite de taille des fichiers	La taille maximale d'un fichier individuel sur les instances de base de données RDS for Oracle est de 16 Tio.	Medium
FTP et SFTP	Amazon RDS ne prend pas en charge FTP ni SFTP.	Medium
Tables partitionnées hybrides	Amazon RDS ne prend pas en charge les tables partitionnées hybrides Oracle.	Élevée
Oracle Data Guard	Amazon RDS ne prend pas en charge Oracle Data Guard pour l'architecture à locataire unique.	Élevée
Oracle Database Vault	Amazon RDS ne prend pas en charge Oracle Database Vault.	Élevée

Limitation	Description	Impact
Coffre-fort de privilèges d'administrateur de base de données Oracle	Amazon RDS a des limitations pour les privilèges d'adminis trateur de base de données Oracle. Pour plus d'informa tions, consultez Limitations des privilèges Oracle DBA.	Élevée
Oracle Enterprise Manager	Amazon RDS ne prend pas en charge Oracle Enterprise Manager pour l'architecture à locataire unique.	Élevée
Oracle Enterprise Manager Agent	Amazon RDS ne prend pas en charge Oracle Enterprise Manager Agent pour l'archite cture à locataire unique.	Medium
Oracle Enterprise Manager Cloud Control Management Repository	Vous ne pouvez pas utiliser une instance de base de données Amazon RDS for Oracle pour le référentiel de gestion de contrôle cloud d'Oracle Enterprise Manager.	Élevée
Base de données Oracle Flashback	Amazon RDS Oracle ne prend pas en charge la fonctionnalité de base de données Oracle Flashback.	Élevée

Limitation	Description	Impact
Oracle Label Security	Amazon RDS ne prend pas en charge Oracle Label Security pour l'architecture à locataire unique. Vous pouvez utiliser Oracle Label Security uniquement avec des bases de données de conteneurs (Oracle CDB) multilocataires.	Élevée
Oracle Messaging Gateway	Amazon RDS ne prend pas en charge Oracle Messaging Gateway.	Élevée
Oracle Real Application Clusters	Amazon RDS ne prend pas en charge Oracle Real Application Clusters (Oracle RAC).	Élevée
Oracle Real Application Testing	Amazon RDS ne prend pas en charge Oracle Real Application Testing.	Élevée
Bases de données Oracle Snapshot Standby	Amazon RDS ne prend pas en charge les bases de données Oracle Snapshot Standby.	Élevée
Synonymes publics	Amazon RDS ne prend pas en charge les synonymes publics des schémas fournis par Oracle.	Medium
Schémas des fonctions non prises en charge	Amazon RDS ne prend pas en charge les schémas pour les fonctionnalités et composants Oracle qui nécessitent des privilèges système.	Élevée

Limitation	Description	Impact
Audit unifié pur	Amazon RDS ne prend pas en charge les audits unifiés purs. Vous pouvez utiliser l'audit unifié en mode mixte.	Medium
Gestionnaire de l'espace de travail	Amazon RDS ne prend pas en charge le schéma `WMSYS` du gestionnaire de l'espace de travail Oracle Database.	Élevée

Le tableau suivant inclut les fonctionnalités de base de données PostgreSQL qu'Amazon RDS for PostgreSQL ne prend pas en charge.

Limitation	Description	Impact
Connexions simultanées	Le nombre maximal de connexions simultanées à votre instance RDS for PostgreSQL est limité par le paramètre `max_connections` .	Faible
Versions les plus récentes	Amazon RDS n'applique pas automatiquement les mises à niveau des versions majeures. Pour effectuer une mise à niveau de version majeure, modifiez votre instance de base de données manuellement. Pour plus d'informations, consultez Choix d'une mise à niveau de version majeure pour PostgreSQL.	Faible

Limitation	Description	Impact
Connexions réservées	Amazon RDS réserve jusqu'à 3 connexions pour la maintenance du système. Si vous spécifiez une valeur pour le paramètre des connexions utilisateur, vous devez ajouter 3 au nombre de connexions que vous envisagez d'utiliser.	Faible
Extensions prises en charge	RDS for PostgreSQL prend en charge un nombre limité d'extensions pour le moteur de base de données PostgreSQL. Vous trouverez la liste des extensions prises en charge dans le groupe de paramètres de base de données par défaut pour votre version de PostgreSQL. Vous pouvez également consulter la liste actuelle des extensions utilisant `psql` en affichant le paramètre `rds.extensions` .	Faible
Fractionnement ou isolation des espaces de table	Vous ne pouvez pas utiliser d'espaces de table pour la répartition ou l'isolement des E/S. Dans RDS for PostgreSQL, tout le stockage se trouve sur un seul volume logique.	Faible

Le tableau suivant inclut les fonctionnalités de base de données SQL Server qu'Amazon RDS for SQL Server ne prend pas en charge.

Limitation	Description	Impact
Sauvegarde dans Microsoft Azure Blob Storage	RDS for SQL Server ne prend pas en charge la sauvegarde vers Microsoft Azure Blob Storage.	Medium
Extension du pool de mémoires tampons	RDS for SQL Server ne prend pas en charge l'extension du pool de mémoires tampons.	Élevée
Politiques de mots de passe personnalisées	RDS for SQL Server ne prend pas en charge les politiques de mot de passe personnalisées.	Medium
Data Quality Services	RDS for SQL Server ne prend pas en charge SQL Server Data Quality Services (DQS).	Élevée
Copie des journaux de transaction de base de données	RDS for SQL Server ne prend pas en charge l'expédition des journaux de base de données.	Élevée
Noms de base de données	Les noms de base de données présentent les limitations suivantes : ils ne peuvent pas commencer par rdsadmin ; ils ne peuvent pas commencer ni se terminer par un espace ou une tabulation ; ils ne peuvent contenir aucun des caractères pouvant créer une nouvelle ligne ; ils ne peuvent	Medium

Limitation	Description	Impact
	pas contenir de guillemets simples (').	
Instantanés de base de données	RDS for SQL Server ne prend pas en charge les instantanés de base de données. Vous ne pouvez utiliser que des instantanés d'instance de base de données dans Amazon RDS.	Medium
Procédures stockées étendues	RDS for SQL Server ne prend pas en charge les procédures stockées étendues, y compris `xp_cmdshell` .	Élevée
Tables de fichiers	RDS for SQL Server ne prend pas en charge les tables de fichiers.	Medium
Support FILESTREAM	RDS for SQL Server n'assure pas la prise en charge de FILESTREAM.	Medium
Serveurs liés	RDS pour SQL Server fournit une prise en charge limitée pour les serveurs liés.	Élevée
Services de machine learning et services R	RDS for SQL Server ne prend pas en charge les services de machine learning et R car vous avez besoin d'un accès au système d'exploitation pour installer ces services.	Élevée

Limitation	Description	Impact
Plans de maintenance	RDS for SQL Server ne prend pas en charge les plans de maintenance.	Élevée
Performance Data Collector	RDS for SQL Server ne prend pas en charge le collecteur de données de performance.	Élevée
Gestion basée sur la politique	RDS for SQL Server ne prend pas en charge la gestion basée sur une politique.	Medium
PolyBase	RDS pour SQL Server n'est pas pris en charge PolyBase.	Élevée
Réplication	RDS for SQL Server ne prend pas en charge la réplication.	Medium
Resource Governor	RDS for SQL Server ne prend pas en charge Resource Governor.	Élevée
Déclencheurs de niveau serveur	RDS for SQL Server ne prend pas en charge les déclencheurs de niveau serveur.	Medium
Points de terminaison Service Broker	RDS for SQL Server ne prend pas en charge les points de terminaison Service Broker.	Élevée
SSAS	Tenez compte des limitations qui s'appliquent à l'exécution de SQL Server Analysis Services (SSAS) sur RDS for SQL Server. Pour plus d'informations, consultez Limites.	Faible

Limitation	Description	Impact
SSIS	Tenez compte des limitations qui s'appliquent à l'exécution de SQL Server Integration Services (SSIS) sur RDS for SQL Server. Pour plus d'informations, consultez Limites.	Faible
SSRS	Tenez compte des limitations qui s'appliquent à l'exécution de SQL Server Reporting Services (SSRS) sur RDS for SQL Server. Pour plus d'informations, consultez Limites.	Faible
Taille de stockage des instances de base de données SQL Server	La taille de stockage maximale pour le stockage à usage général (SSD) SQL Server et les instances de stockage IOPS provisionnées est de 16 Tio. La taille de stockage maximale pour les instances de stockage magnétique SQL Server est de 1 Tio.	Élevée
Stretch Database	RDS for SQL Server ne prend pas en charge la fonctionnalité SQL Server Stretch Database.	Medium
Points de terminaison T-SQL	RDS for SQL Server ne prend pas en charge toutes les opérations qui utilisent `CREATE ENDPOINT`.	Élevée

Limitation	Description	Impact
Propriété de base de données TRUSTWORTHY	RDS for SQL Server ne prend pas en charge la propriété de base de données `TRUSTWORTHY` , car elle nécessite le rôle `sysadmin`.	Medium

Le tableau suivant contient une liste des problèmes liés aux recommandations. DMS Fleet Advisor analyse les fonctionnalités des bases de données source et cible et indique ces limites de migration. La limitation liée à l'impact du bloqueur signifie que DMS Fleet Advisor ne peut pas générer de recommandations cibles pour la base de données source.

Limitation	Description	Impact
L'instance appropriée n'a pas été trouvée	AWS DMSImpossible de trouver une instance cible pouvant fonctionner comme une destination de migration de taille appropriée pour une combinaison des métriques de votre base de données source.	Bloqueur
L'instance appropriée n'est pas trouvée par IOPS	La base de données source utilise un nombre d'IOPS supérieur au nombre maximal d'IOPS pour les instances de base de données cibles possibles.	Bloqueur
L'instance appropriée n'est pas trouvée par la RAM	La base de données source utilise un certain nombre de Go de RAM, ce qui dépasse la taille maximale de RAM	Bloqueur

Limitation	Description	Impact
	pour les instances de base de données cibles possibles.	
L'instance appropriée n'est pas trouvée en fonction de la taille de stockage	La base de données source utilise un certain nombre de To de stockage, ce qui dépasse la taille de stockage maximale pour les instances de base de données cibles possibles.	Bloqueur
L'instance appropriée n'est pas trouvée par édition	La base de données source possède une édition qui n'est pas prise en charge par Amazon RDS.	Bloqueur
L'instance appropriée n'est pas trouvée par les cœurs du processeur	La base de données source possède un nombre de cœurs de processeur supérieur au nombre maximal de cœurs de processeur pour les instances de base de données cibles possibles.	Bloqueur
L'instance appropriée n'est pas trouvée par version	La version de votre base de données source AWS DMS ne la reconnaît pas.	Bloqueur

Limitation	Description	Impact
Le paramètre du processeur n'est pas défini	Le collecteur de données DMS n'a pas collecté d'informations sur le processeur utilisé par votre base de données source. Assurez-vous d'avoir collecté les métriques requises et d'avoir configuré les informations d'identification pour le transfert de données dans votre collecteur de données. veuillez consulter Configuration des informations d'identification pour le transfert de données.	Bloqueur
Le paramètre de mémoire n'est pas défini	Le collecteur de données DMS n'a pas collecté d'informations sur la mémoire utilisée par votre base de données source. Assurez-vous d'avoir collecté les métriques requises et d'avoir configuré les informations d'identification pour le transfert de données dans votre collecteur de données. veuillez consulter Configuration des informations d'identification pour le transfert de données.	Bloqueur

Limitation	Description	Impact
Le paramètre de taille de stockage n'est pas défini	Le collecteur de données DMS n'a pas collecté d'informations sur la taille de stockage utilisée par votre base de données source. Assurez-vous d'avoir collecté les métriques requises et d'avoir configuré les informations d'identification pour le transfert de données dans votre collecteur de données. veuillez consulter Configuration des informations d'identification pour le transfert de données.	Bloqueur
Le paramètre IOPS de stockage n'est pas défini	Le collecteur de données DMS n'a pas collecté les métriques d'IOPS de stockage pour les utilisations de votre base de données source. Assurez-vous d'avoir collecté les métriques requises et d'avoir configuré les informations d'identification pour le transfert de données dans votre collecteur de données.	Bloqueur

Limitation	Description	Impact
Pas assez de données	Le collecteur de données DMS n'a pas collecté suffisamm ent de données pour générer une recommandation cible. Assurez-vous d'avoir configuré les informations d'identif ication pour le transfert de données dans votre collecteur de données. veuillez consulter Configuration des informations d'identification pour le transfert de données.	Bloqueur
L'édition de la base de données n'est pas définie	Le collecteur de données DMS n'a pas collecté d'informa tions sur l'édition de votre base de données source. Assurez-vous d'avoir collecté les métriques requises et d'avoir configuré les informati ons d'identification pour le transfert de données dans votre collecteur de données. veuillez consulter Configura tion des informations d'identif ication pour le transfert de données.	Bloqueur
Erreur inconnue	DMS Fleet Advisor ne peut pas générer de recommand ations cibles pour votre base de données source.	Bloqueur

Limitation	Description	Impact
La version de la base de données n'est pas définie	Le DMS Fleet Advisor n'a pas collecté d'informations sur la version de votre base de données source. DMS Fleet Advisor vous recommande d'utiliser la dernière version de base de données pour votre base de données source. Si vous choisisse z cette recommandation, vous devez mettre à niveau la version de votre base de données. Passez en revue les recommandations cibles générées pour votre base de données source et assurez-v ous qu'elles répondent à vos exigences.	Élevée
Augmenter le nombre de connexions à la base de données dans les paramètres RDS	Votre base de données source nécessite un certain nombre de connexions. Par défaut, le nombre de connexions disponibles pour les instances de base de données Amazon RDS est différent. Assurez-v ous de modifier cette valeur par défaut lorsque vous créez votre instance de base de données RDS. Pour ce faire, mettez à jour la valeur du `max_connections` paramètre dans les groupes de paramètres.	Medium

Limitation	Description	Impact
L'édition Target est compatible	La recommandation cible pour votre base de données source utilise une édition de base de données différente. L'édition de votre base de données source prend en charge les mêmes fonctionnalités que l'édition cible recommandée. Cependant, le choix de cette nouvelle édition de base de données peut augmenter vos dépenses.	Medium
Le paramètre de débit de stockage n'est pas défini	Le collecteur de données DMS n'a pas collecté les mesures de débit de stockage pour les utilisations de votre base de données source. Passez en revue les recommandations cibles générées pour votre base de données source et assurez-vous qu'elles répondent à vos exigences.	Medium

Limitation	Description	Impact
Le paramètre du numéro de connexion à la base de données n'est pas défini	Le collecteur de données DMS n'a pas collecté d'informations sur le nombre de connexion s utilisées par votre base de données source. Passez en revue les recommandations cibles générées pour votre base de données source et assurez-vous qu'elles répondent à vos exigences . Vous pouvez également demander une augmentation de quota.	Medium
Version rétrogradée de la base de données	Votre base de données source fonctionne sur une version supérieure à celle de la base de données Amazon RDS. Pour rétrograder la version de votre base de données, assurez-vous de ne pas utiliser les fonctionnalités qui ne sont pas implément ées dans la version inférieur e. Vous pouvez également utiliser Amazon EC2 comme cible de migration.	Medium

Limitation	Description	Impact
L'édition cible est différente	La recommandation cible pour votre base de données source utilise une édition de base de données différente. L'édition de votre base de données source est compatible avec l'édition cible recommandée. Toutefois, l'édition de base de données cible recommandée ne prend pas en charge certaines fonctionnalités de l'édition de votre base de données source. Le choix de cette nouvelle édition de base de données peut augmenter vos dépenses.	Medium

Limitation	Description	Impact
Mise à niveau depuis une version non prise en charge	Votre base de données source a atteint la fin de la phase de support. Pour utiliser la dernière version du moteur de base de données comme cible, mettez à niveau votre base de données avant la migration. Vous pouvez également utiliser Amazon EC2 comme cible de migration . En fonction du moteur de base de données, utilisez l'un des liens suivants pour en savoir plus : Mise à niveau de MySQL Mettre à niveau SQL Server Mettre à niveau OracleDB Mettre à niveau PostgreSQL	Medium

Résolution des problèmes liés aux recommandations cibles

Dans la liste suivante, vous trouverez les actions à entreprendre lorsque vous rencontrez des problèmes avec la fonctionnalité Recommandations cibles de DMS Fleet Advisor.

Rubriques

- Je ne peux pas voir d'estimations de prix pour les recommandations cibles
- Je ne peux pas voir les graphiques d'utilisation des ressources
- Je ne vois pas le statut de collecte des métriques

Je ne peux pas voir d'estimations de prix pour les recommandations cibles

Si vous voyez Aucune donnée pour Coût mensuel estimé pour une recommandation dont le statut est Réussite, assurez-vous d'avoir autorisé votre utilisateur IAM à accéder à l'API AWS Price List Service. Pour ce faire, vous devez créer la politique qui inclut l'autorisation `pricing:GetProducts` et l'ajouter à votre utilisateur IAM comme décrit dans Création de ressources IAM.

DMS Fleet Advisor ne calcule pas le coût mensuel estimé pour les recommandations dont le statut est Échec.

Je ne peux pas voir les graphiques d'utilisation des ressources

Si le message « Impossible de charger les métriques » s'affiche une fois que vous avez développé la section Utilisation et capacité de la source, assurez-vous d'avoir autorisé votre utilisateur IAM à consulter les CloudWatch tableaux de bord Amazon. Pour ce faire, vous devez ajouter la politique requise à votre utilisateur IAM comme décrit dans Création de ressources IAM.

Vous pouvez également créer une politique personnalisée qui inclut les autorisations `cloudwatch:GetDashboard`, `cloudwatch:ListDashboards`, `cloudwatch:PutDashboard` et `cloudwatch:DeleteDashboards`. Pour plus d'informations, consultez la section Utilisation CloudWatch des tableaux de bord Amazon dans le guide de CloudWatch l'utilisateur Amazon.

Je ne vois pas le statut de collecte des métriques

Si vous voyez Aucune donnée disponible pour Collecte de métriques lorsque vous choisissez Générer des recommandations, assurez-vous d'avoir collecté des données. Pour plus d'informations, consultez Collecte de données pour AWS DMS Fleet Advisor.

Si vous rencontrez ce problème après avoir collecté des données, assurez-vous d'avoir autorisé votre utilisateur IAM à accéder à Amazon CloudWatch. `cloudwatch:Get*` DMS Fleet Advisor utilise un rôle lié à un service pour publier en votre nom les indicateurs de performance de la CloudWatch base de données collectés. Assurez-vous de créer un rôle lié à un service à utiliser avec DMS Fleet Advisor. Pour plus d'informations, consultez Création de ressources IAM.

Limites de DMS Fleet Advisor

Les limites liées à l'utilisation du DMS Fleet Advisor sont les suivantes :

- DMS Fleet Advisor génère one-to-one des recommandations. Pour chaque base de données source, DMS Fleet Advisor détermine un moteur cible unique. DMS Fleet Advisor ne gère pas les

serveurs multilocataires et ne fournit pas de recommandations pour exécuter plusieurs bases de données sur une seule instance de base de données cible.

- DMS Fleet Advisor ne fournit aucune recommandation concernant les mises à niveau des versions de base de données disponibles.
- DMS Fleet Advisor génère des recommandations pour un maximum de 100 bases de données à la fois.
- Si vous installez le collecteur de données DMS, qui est une application Windows, assurez-vous d'installer également .NET Framework 4.8 et PowerShell 6.0 et versions ultérieures. Pour connaître la configuration matérielle requise, consultez Installation d'un collecteur de données.
- Le collecteur de données DMS a besoin d'autorisations pour exécuter des demandes via le protocole LDAP sur votre serveur de domaine.
- Le collecteur de données DMS nécessite le script sudo SSH exécuté sous Linux.
- Le collecteur de données DMS nécessite des autorisations pour exécuter à distance PowerShell, Windows Management Instrumentation (WMI), WMI Query Language (WQL) et des scripts de registre sous Windows.
- Pour MySQL et PostgreSQL, DMS Fleet Advisor ne peut pas collecter de métriques de performance à partir de la base de données. Au lieu de cela, DMS Fleet Advisor collecte les métriques de serveur OS. Par conséquent, vous ne pouvez pas générer de recommandations basées sur des métriques d'utilisation pour les bases de données MySQL et PostgreSQL exécutées sur Amazon RDS et Aurora.

Conversion de schémas de base de données à l'aide de la conversion de schéma DMS

La conversion du schéma DMS in AWS Database Migration Service (AWS DMS) rend les migrations de bases de données entre différents types de bases de données plus prévisibles. Utilisez la conversion de schéma DMS pour évaluer la complexité de votre migration pour votre fournisseur de données source, puis pour convertir les schémas de base de données et les objets de code. Vous pouvez alors appliquer le code converti à la base de données cible.

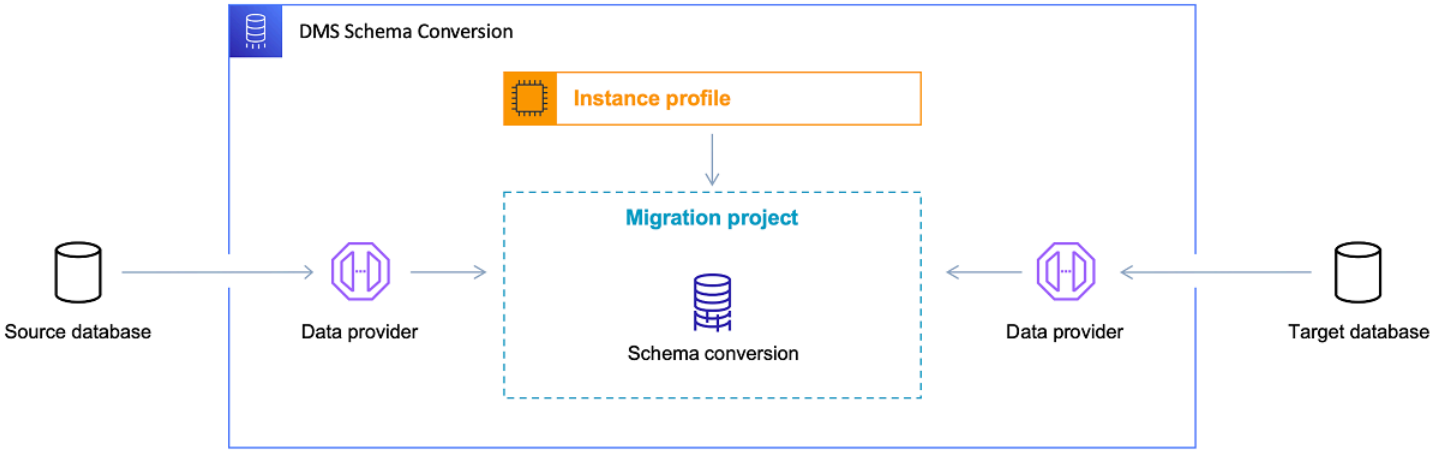
La conversion de schéma DMS convertit automatiquement vos schémas de base de données source et la plupart des objets de code de base de données dans un format compatible avec la base de données cible. Cette conversion inclut les tables, les vues, les procédures stockées, les fonctions, les types de données, les synonymes, etc. Tous les objets que la conversion de schéma DMS ne peut pas convertir automatiquement sont clairement marqués. Pour terminer la migration, vous pouvez convertir ces objets manuellement.

De manière générale, la conversion de schéma DMS fonctionne avec les trois composants suivants : les profils d'instance, les fournisseurs de données et les projets de migration. Un profil d'instance spécifie les paramètres réseau et de sécurité. Un fournisseur de données stocke les informations d'identification de connexion à la base de données. Un projet de migration contient des fournisseurs de données, un profil d'instance et des règles de migration. AWS DMS utilise des fournisseurs de données et un profil d'instance pour concevoir un processus de conversion des schémas de base de données et des objets de code.

Pour obtenir la liste des bases de données sources prises en charge, consultez Sources pour la conversion de schéma DMS.

Pour obtenir la liste des bases de données cibles prises en charge, consultez Cibles pour la conversion de schéma DMS.

Le diagramme suivant illustre le processus de conversion de schéma DMS.



Consultez les rubriques suivantes pour mieux comprendre comment utiliser la conversion de schéma DMS.

Rubriques

- Soutenu Régions AWS
- Fonctionnalités de conversion de schéma
- Limitation de la conversion de schéma
- Bien démarrer avec la conversion de schéma DMS
- Configuration d'un réseau pour la conversion de schéma DMS
- Création de fournisseurs de données sources dans la conversion de schéma DMS
- Création de fournisseurs de données cibles dans la conversion de schéma DMS
- Gestion de projets de migration pour la conversion de schéma DMS
- Création de rapports d'évaluation de migration de base de données avec la conversion de schéma DMS
- Utilisation de la conversion de schéma DMS
- Utilisation de packs d'extension dans la conversion de schéma DMS

Soutenu Régions AWS

Vous pouvez créer un projet de migration de schéma DMS Conversion comme suit Régions AWS. Dans d'autres régions, vous pouvez utiliser l' AWS Schema Conversion Tool. Pour plus d'informations à ce sujet AWS SCT, consultez le AWS Schema Conversion Tool User Guide.

Nom de la région	Région
US East (Virginie du Nord)	us-east-1
USA Est (Ohio)	us-east-2
USA Ouest (Oregon)	us-west-2
Asie-Pacifique (Tokyo)	ap-northeast-1
Asie-Pacifique (Singapour)	ap-southeast-1
Asie-Pacifique (Sydney)	ap-southeast-2
Europe (Francfort)	eu-central-1
Europe (Stockholm)	eu-north-1
Europe (Irlande)	eu-west-1

Fonctionnalités de conversion de schéma

La conversion de schéma DMS fournit les fonctionnalités suivantes :

- DMS Schema Conversion gère automatiquement les AWS Cloud ressources requises pour votre projet de migration de base de données. Ces ressources incluent les profils d'instance, les fournisseurs de données et AWS Secrets Manager les secrets. Ils incluent également les rôles AWS Identity and Access Management (IAM), les compartiments Amazon S3 et les projets de migration.
- Vous pouvez utiliser la conversion de schéma DMS pour vous connecter à la base de données source, lire les métadonnées et créer des rapports d'évaluation de migration de base de données. Vous pouvez ensuite enregistrer le rapport dans un compartiment Amazon S3. Ces rapports fournissent un résumé de vos tâches de conversion de schéma et les détails des éléments que la conversion de schéma DMS ne peut pas convertir automatiquement vers la base de données cible. Les rapports d'évaluation de migration de base de données permettent d'évaluer la part de votre projet de migration que la conversion de schéma DMS peut automatiser. De plus, ces rapports permettent d'estimer l'effort manuel requis pour effectuer la conversion. Pour de plus

amples informations, veuillez consulter Création de rapports d'évaluation de migration de base de données avec la conversion de schéma DMS.

- Une fois que vous êtes connecté à vos fournisseurs de données source et cible, la conversion de schéma DMS peut convertir vos schémas de base de données source existants vers le moteur de base de données cible. Vous pouvez choisir n'importe quel élément de schéma de votre base de données source de convertir. Après avoir converti le code de base de données via la conversion de schéma DMS, vous pouvez passer en revue le code source et le code converti. Vous pouvez également enregistrer le code SQL converti dans un compartiment Amazon S3.
- Avant de convertir les schémas de base de données source, vous pouvez configurer des règles de transformation. Vous pouvez utiliser des règles de transformation pour changer le type de données des colonnes, déplacer des objets d'un schéma à un autre et modifier les noms des objets. Vous pouvez appliquer des règles de transformation aux bases de données, aux schémas, aux tables et aux colonnes. Pour de plus amples informations, veuillez consulter Configuration des règles de transformation.
- Vous pouvez modifier les paramètres de conversion pour améliorer les performances du code converti. Ces paramètres sont spécifiques à chaque paire de conversion et dépendent des fonctionnalités de la base de données source que vous utilisez dans votre code. Pour de plus amples informations, veuillez consulter Spécification des paramètres de conversion de schéma.
- Dans certains cas, la conversion de schéma DMS ne peut pas convertir les fonctionnalités de base de données source en fonctionnalités Amazon RDS équivalentes. Dans ces cas, la conversion de schéma DMS crée un pack d'extension dans la base de données cible afin d'émuler les fonctionnalités qui n'ont pas été converties. Pour de plus amples informations, veuillez consulter Utilisation de packs d'extension.
- Vous pouvez appliquer le code converti et le schéma de kit d'extension à la base de données cible. Pour de plus amples informations, veuillez consulter Application de votre code converti.
- La conversion de schéma DMS prend en charge toutes les fonctionnalités de la dernière AWS SCT version. Pour plus d'informations, consultez les dernières notes de publication de AWS SCT.
- Vous pouvez modifier le code SQL converti avant que DMS ne le migre vers la base de données cible. Pour de plus amples informations, veuillez consulter Modification et enregistrement de votre code SQL converti .

Limitation de la conversion de schéma

La conversion du schéma DMS est une version Web du AWS Schema Conversion Tool ()AWS SCT. La conversion de schéma DMS prend en charge moins de plateformes de base de données et fournit

des fonctionnalités plus limitées par rapport à l'application de bureau AWS SCT . Pour convertir des schémas d'entrepôt de données, des infrastructures de big data, du code SQL d'application et des processus ETL, utilisez AWS SCT. Pour plus d'informations à ce sujet AWS SCT, consultez le AWS Schema Conversion Tool User Guide.

Les limitations suivantes s'appliquent lorsque vous utilisez la conversion de schéma DMS pour la conversion de schéma de base de données :

- Vous ne pouvez pas enregistrer un projet de migration et l'utiliser en mode hors connexion.
- Vous ne pouvez pas modifier le code SQL de la source dans un projet de migration pour la conversion de schéma DMS. Pour modifier le code SQL de la base de données source, utilisez votre éditeur SQL habituel. Choisissez Actualiser à partir de la base de données pour ajouter le code mis à jour dans votre projet de migration.
- Les règles de migration dans la conversion de schéma DMS ne permettent pas de modifier le classement des colonnes. Vous ne pouvez pas non plus utiliser les règles de migration pour déplacer des objets vers un nouveau schéma.
- Vous ne pouvez pas appliquer de filtres aux arborescences des bases de données source et cible pour afficher uniquement les objets de base de données qui répondent à la clause de filtrage.
- Le pack d'extension DMS Schema Conversion n'inclut pas AWS Lambda les fonctions qui émulent l'envoi d'e-mails, la planification des tâches et d'autres fonctionnalités de votre code converti.
- La conversion du schéma DMS n'utilise pas de clés KMS gérées par le client pour accéder aux ressources du client. AWS Par exemple, la conversion de schéma DMS ne prend pas en charge l'utilisation d'une clé KMS gérée par le client pour accéder aux données client dans Amazon S3.

Bien démarrer avec la conversion de schéma DMS

Pour bien démarrer avec la conversion de schéma DMS, consultez le didacticiel suivant. Vous y apprendrez à configurer la conversion de schéma DMS, à créer un projet de migration et à vous connecter à vos fournisseurs de données. Vous pourrez ensuite apprendre à évaluer la complexité de votre migration et à convertir la base de données source dans un format compatible avec la base de données cible. De plus, vous pourrez apprendre à appliquer le code converti à la base de données cible.

Le didacticiel suivant couvre les tâches prérequises et explique la conversion d'une base de données Amazon RDS for SQL Server vers Amazon RDS for MySQL. Vous pouvez utiliser l'un quelconque

des fournisseurs de données sources et cibles pris en charge. Pour de plus amples informations, veuillez consulter Fournisseurs de données sources pour la conversion de schéma DMS.

Pour plus d'informations sur la conversion du schéma DMS, consultez les guides de step-by-step migration d'Oracle vers PostgreSQL et de SQL Server vers MySQL.

Cette vidéo présente l'interface utilisateur de conversion de schéma DMS et vous aide à vous familiariser avec les principaux composants de ce service.

Rubriques

- Conditions préalables à l'utilisation de la conversion de schéma DMS
- Étape 1 : Créer un profil d'instance
- Étape 2 : Configurer vos fournisseurs de données
- Étape 3 : Créer un projet de migration
- Étape 4 : Créer un rapport d'évaluation
- Étape 5 : Convertir le code source
- Étape 6 : Appliquer le code converti
- Étape 7 : Nettoyer et résoudre les problèmes

Conditions préalables à l'utilisation de la conversion de schéma DMS

Pour configurer la conversion de schéma DMS, exécutez les tâches suivantes. Vous pouvez ensuite configurer un profil d'instance, ajouter des fournisseurs de données et créer un projet de migration.

Rubriques

- Création d'un VPC basé sur Amazon VPC
- Créer un compartiment Amazon S3
- Stockez les identifiants de base de données dans AWS Secrets Manager
- Création de rôles IAM

Création d'un VPC basé sur Amazon VPC

Au cours de cette étape, vous créez un cloud privé virtuel (VPC) dans votre. Compte AWS Ce VPC est basé sur le service Amazon Virtual Private Cloud (Amazon VPC) et contient vos ressources. AWS

Pour créer un VPC pour la conversion de schéma DMS

1. Connectez-vous à la console Amazon VPC AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/vpc/.
2. Sélectionnez Create VPC (Créer un VPC).
3. Sur la page Créer un VPC, entrez les paramètres suivants :
   - Ressources à créer : VPC et plus encore
   - Génération automatique d'identifications de noms : choisissez Générer automatiquement et entrez un nom unique global. Par exemple, saisissez **`sc-vpc`**.
   - IPv4 CIDR block (Bloc d'adresse CIDR IPv4) : **`10.0.1.0/24`**
   - Passerelles NAT : Dans 1 AZ
   - VPC endpoints (Points de terminaison VPC) : None (Aucun)
4. Conservez le reste des paramètres tels qu'ils sont, puis choisissez Créer un VPC.
5. Choisissez Sous-réseaux et notez vos identifiants de sous-réseaux publics et privés.

   Pour vous connecter à vos bases de données Amazon RDS, créez un groupe de sous-réseaux incluant des sous-réseaux publics.

   Pour vous connecter à vos bases de données sur site, créez un groupe de sous-réseaux qui inclut des sous-réseaux privés. Pour de plus amples informations, veuillez consulter Étape 1 : Créer un profil d'instance.
6. Choisissez Passerelle NAT. Choisissez votre passerelle NAT et notez votre adresse IP Elastic.

   Configurez votre réseau pour vous assurer qu'il AWS DMS peut accéder à votre base de données source sur site à partir de l'adresse IP publique de cette passerelle NAT. Pour de plus amples informations, veuillez consulter Utilisation d'une connexion Internet à un VPC.

Utilisez ce VPC lorsque vous créez votre profil d'instance et vos bases de données cibles sur Amazon RDS.

Créer un compartiment Amazon S3

Pour stocker les informations de votre projet de migration, créez un compartiment Amazon S3. La conversion de schéma DMS utilise ce compartiment Amazon S3 pour enregistrer des éléments tels que des rapports d'évaluation, du code SQL converti, des informations sur des objets de schéma de base de données, etc.

Pour créer un compartiment Amazon S3 pour la conversion de schéma DMS

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Choisissez Créer un compartiment.
3. Sur la page Créer un compartiment, sélectionnez un nom unique global pour votre compartiment S3. Par exemple, saisissez **`sc-s3-bucket`**.
4. Pour Région AWS, choisissez votre région.
5. Pour Gestion des versions de compartiment, choisissez Activer.
6. Conservez les autres paramètres tels quels, puis choisissez Créer un compartiment.

Stockez les identifiants de base de données dans AWS Secrets Manager

Stockez vos informations d'identification de base de données source et cible dans AWS Secrets Manager. Assurez-vous de reproduire ces secrets dans votre Région AWS. La conversion de schéma DMS utilise ces secrets pour se connecter à vos bases de données dans le cadre du projet de migration.

Pour stocker les informations d'identification de votre base de données dans AWS Secrets Manager

1. Connectez-vous à la AWS Secrets Manager console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/secretsmanager/.
2. Choisissez Store a new secret (Stocker un nouveau secret).
3. La page Choisir un type de secret s'ouvre. Pour Secret type (Type de secret), choisissez le type d'informations d'identification de base de données à stocker :

   - Informations d'identification pour une base de données Amazon RDS : choisissez cette option pour stocker les informations d'identification pour la base de données Amazon RDS. Dans Informations d'identification, entrez les informations d'identification pour la base de données. Pour Database (Base de données), choisissez votre base de données.
   - Informations d'identification pour une autre base de données : choisissez cette option pour stocker les informations d'identification pour vos bases de données Oracle ou SQL Server sources. Dans Informations d'identification, entrez les informations d'identification pour la base de données.
   - Autre type de secret : choisissez cette option pour stocker uniquement le nom d'utilisateur et le mot de passe pour vous connecter à la base de données. Choisissez Ajouter une ligne pour

ajouter deux paires clé-valeur. Assurez-vous d'utiliser **`username`** et **`password`** pour les noms de clé. Pour les valeurs associées à ces clés, entrez les informations d'identification pour la base de données.

4. Pour Clé de chiffrement, choisissez la AWS KMS clé que Secrets Manager utilise pour chiffrer la valeur secrète. Choisissez Suivant.
5. Sur la page Configurer le secret, entrez un nom de secret descriptif. Par exemple, entrez **`sc-source-secret`** ou **`sc-target-secret`**.
6. Choisissez Répliquer le secret, puis, dans Région AWS, choisissez votre région. Choisissez Suivant.
7. Sur la page Configurer la rotation, choisissez Suivant.
8. Dans la page Review (Révision), passez en revue vos paramètres, puis choisissez Store (Stocker).

Pour stocker les informations d'identification pour vos bases de données sources et cibles, répétez ces étapes.

Création de rôles IAM

Créez des rôles AWS Identity and Access Management (IAM) à utiliser dans votre projet de migration. La conversion de schéma DMS utilise ces rôles IAM pour accéder à votre compartiment Amazon S3 et aux informations d'identification de base de données stockées dans AWS Secrets Manager.

Pour créer un rôle IAM qui donne accès à votre compartiment Amazon S3

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Roles (Rôles).
3. Sélectionnez Create role (Créer un rôle).
4. Sur la page Sélectionner une entité de confiance, choisissez Service AWS . Choisissez DMS.
5. Choisissez Suivant. La page Ajouter des autorisations s'ouvre.
6. Pour Politiques de filtre, entrez **`S3`**. Choisissez Amazon S3 FullAccess.
7. Choisissez Suivant. La page Nommer, vérifier et créer s'ouvre.
8. Pour Nom du rôle, entrez un nom descriptif. Par exemple, saisissez **`sc-s3-role`**. Sélectionnez Créer un rôle.

9. Sur la page Rôles, entrez **`sc-s3-role`** pour Nom du rôle. Choisissez sc-s3-role.
10. Sur la page sc-s3-role, choisissez l'onglet Relations d'approbation. Choisissez Edit trust policy (Modifier la politique d'approbation).
11. Sur la page Modifier la politique d'approbation, modifiez les relations d'approbation du rôle afin d'utiliser le principal de service `schema-conversion.dms.amazonaws.com` comme entité de confiance.
12. Choisissez Update Trust Policy (Mettre à jour la stratégie d'approbation).

Pour créer un rôle IAM donnant accès à AWS Secrets Manager

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Roles (Rôles).
3. Sélectionnez Create role (Créer un rôle).
4. Sur la page Sélectionner une entité de confiance, choisissez Service AWS . Choisissez DMS.
5. Choisissez Suivant. La page Ajouter des autorisations s'ouvre.
6. Pour Politiques de filtre, entrez **`Secret`**. Choisissez SecretsManagerReadWrite.
7. Choisissez Suivant. La page Nommer, vérifier et créer s'ouvre.
8. Pour Nom du rôle, entrez un nom descriptif. Par exemple, saisissez **`sc-secrets-manager-role`**. Sélectionnez Créer un rôle.
9. Sur la page Rôles, entrez **`sc-secrets-manager-role`** pour Nom du rôle. Choisissez sc-secrets-manager-role.
10. Sur la sc-secrets-manager-rolepage, choisissez l'onglet Relations de confiance. Choisissez Edit trust policy (Modifier la politique d'approbation).
11. Sur la page Modifier la politique de confiance, modifiez les relations de confiance pour le rôle à utiliser `schema-conversion.dms.amazonaws.com` et pour votre directeur de service AWS DMS régional en tant qu'entités de confiance. Ce directeur de service AWS DMS régional a le format suivant.

    ```
    dms.region-name.amazonaws.com
    ```

    Remplacez *`region-name`* par le nom de votre région, tel que `us-east-1`.

    L'exemple de code suivant montre le principal pour la région `us-east-1`.

```
dms.us-east-1.amazonaws.com
```

L'exemple de code suivant montre une politique de confiance pour accéder à la conversion de AWS DMS schéma.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "dms.us-east-1.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "schema-conversion.dms.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

12. Choisissez Update Trust Policy (Mettre à jour la stratégie d'approbation).

Étape 1 : Créer un profil d'instance

Avant de créer un profil d'instance, configurez un groupe de sous-réseaux pour votre profil d'instance. Pour plus d'informations sur la création d'un groupe de sous-réseaux pour votre projet de AWS DMS migration, consultezCréation d'un groupe de sous-réseaux.

Vous pouvez créer un profil d'instance comme le décrit la procédure suivante. Dans ce profil d'instance, vous spécifiez les paramètres réseau et de sécurité pour votre projet de conversion de schéma DMS.

Pour créer un profil d'instance

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Dans le volet de navigation, choisissez Profils d'instance, puis Créer un profil d'instance.
3. Pour Nom, entrez un nom unique pour le profil d'instance. Par exemple, saisissez **`sc-instance`**.
4. Pour Type de réseau, choisissez IPv4 pour créer un profil d'instance prenant en charge l'adressage IPv4 uniquement. Pour créer un profil d'instance prenant en charge l'adressage IPv4 et IPv6, choisissez Mode double pile.
5. Pour Cloud privé virtuel (VPC), choisissez le VPC que vous avez créé à l'étape des conditions préalables requises.
6. Pour Groupe de sous-réseaux, choisissez le groupe de sous-réseaux pour votre profil d'instance. Pour vous connecter aux bases de données Amazon RDS, utilisez un groupe de sous-réseaux qui inclut des sous-réseaux publics. Pour vous connecter aux bases de données sur site, utilisez un groupe de sous-réseaux qui inclut des sous-réseaux privés.
7. Choisissez Créer un profil d'instance.

Pour créer un projet de migration, utilisez ce profil d'instance.

Étape 2 : Configurer vos fournisseurs de données

Ensuite, vous devez créer les fournisseurs de données qui décrivent vos bases de données source et cible. Pour chaque fournisseur de données, vous spécifiez un type de magasin de données et des informations d'emplacement. Vous ne stockez pas vos informations d'identification pour une base de données dans un fournisseur de données.

Pour créer un fournisseur de données pour une base de données source sur site

1. Connectez-vous à la AWS Management Console AWS DMS console et ouvrez-la.
2. Dans le volet de navigation, choisissez Fournisseurs d'identité, puis Créer un fournisseur de données.
3. Pour Nom, entrez un nom unique pour le fournisseur de données source. Par exemple, saisissez **`sc-source`**.
4. Pour Type de moteur, choisissez le type de moteur de base de données pour votre fournisseur de données.

5. Fournissez vos informations de connexion à la base de données source. Les paramètres de connexion dépendent de votre moteur de base de données source. Pour de plus amples informations, veuillez consulter Création de fournisseurs de données.
6. Pour Mode SSL (Secure Socket Layer), choisissez le type de mise en œuvre du protocole SSL.
7. Choisissez Créer un fournisseur de données.

Pour créer un fournisseur de données pour une base de données Amazon RDS cible

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la.
2. Dans le volet de navigation, choisissez Fournisseurs d'identité, puis Créer un fournisseur de données.
3. Pour Configuration, choisissez Instance de base de données RDS.
4. Pour Base de données RDS, choisissez Parcourir et choisissez la base de données. La conversion de schéma DMS récupère automatiquement les informations relatives au type de moteur, au nom de serveur et au port.
5. Pour Nom, entrez un nom unique pour le fournisseur de données cible. Par exemple, saisissez **`sc-target`**.
6. Pour Database name (Nom de la base de données), entrez le nom de votre base de données.
7. Pour Mode SSL (Secure Socket Layer), choisissez le type de mise en œuvre du protocole SSL.
8. Choisissez Créer un fournisseur de données.

Étape 3 : Créer un projet de migration

Vous pouvez maintenant créer un projet de migration. Dans ce projet de migration, vous spécifiez vos fournisseurs de données source et cible, ainsi que votre profil d'instance.

Pour créer un projet de migration

1. Choisissez Projets de migration, puis choisissez Créer un projet de migration.
2. Pour Nom, entrez un nom unique pour votre projet de migration. Par exemple, saisissez **`sc-project`**.
3. Pour Profil d'instance, choisissez **`sc-instance`**.
4. Pour Source, choisissez Parcourir, puis choisissez **`sc-source`**.
5. Pour ID secret, choisissez **`sc-source-secret`**.

6. Pour Rôle IAM, choisissez **sc-secrets-manager-role**.
7. Pour Cible, choisissez Parcourir, puis choisissez **sc-target**.
8. Pour ID secret, choisissez **sc-target-secret**.
9. Pour Rôle IAM, choisissez **schema-conversion-role**.
10. Choisissez Créer un projet de migration.

Étape 4 : Créer un rapport d'évaluation

Pour évaluer la complexité de la migration, créez le rapport d'évaluation de la migration de base de données. Ce rapport inclut la liste de tous les objets de base de données que la conversion de schéma DMS ne peut pas convertir automatiquement.

Pour créer un rapport d'évaluation

1. Choisissez Projets de migration, puis choisissez **sc-project**.
2. Choisissez Conversion de schéma, puis choisissez Lancer la conversion de schéma.
3. Dans le volet de la base de données source, choisissez le schéma de base de données à évaluer. Cochez également la case correspondant au nom de ce schéma.
4. Dans le volet de la base de données source, choisissez Évaluer dans le menu Actions. La boîte de dialogue Évaluer s'affiche.
5. Choisissez Évaluer dans cette boîte de dialogue pour confirmer votre choix.

   L'onglet Résumé indique le nombre d'éléments que la conversion de schéma DMS peut convertir automatiquement pour les objets de stockage de base de données et les objets de code de base de données.
6. Choisissez Éléments d'action pour afficher la liste de tous les objets de base de données que la conversion de schéma DMS ne peut pas convertir automatiquement. Passez en revue les actions recommandées pour chaque élément.
7. Pour enregistrer une copie de votre rapport d'évaluation, choisissez Exporter les résultats. Choisissez ensuite l'un des formats suivants : CSV ou PDF. La boîte de dialogue Exporter apparaît.
8. Choisissez Exporter pour confirmer votre choix.
9. Choisissez Compartiment S3. La console Amazon S3 s'ouvre.
10. Choisissez Télécharger pour enregistrer votre rapport d'évaluation.

Étape 5 : Convertir le code source

Vous pouvez convertir votre schéma de base de données source à l'aide de la procédure suivante. Vous pouvez alors enregistrer le code converti sous forme de scripts SQL dans un fichier texte.

Pour convertir votre schéma de base de données

1. Dans le volet de la base de données source, choisissez le schéma de base de données à convertir. Cochez également la case correspondant au nom de ce schéma.
2. Dans le volet de la base de données source, choisissez Convertir dans le menu Actions. La boîte de dialogue Convertir apparaît.
3. Choisissez Convertir dans cette boîte de dialogue pour confirmer votre choix.
4. Choisissez un objet de base de données dans le volet de base de données source. La conversion de schéma DMS affiche le code source et le code converti pour cet objet.
5. Dans le volet de la base de données cible, choisissez le schéma de base de données converti. Cochez également la case correspondant au nom de ce schéma.
6. Pour Actions, choisissez Enregistrer en tant que SQL. La boîte de dialogue Enregistrer s'affiche.
7. Choisissez Enregistrer en tant que SQL pour confirmer votre choix.
8. Choisissez Compartiment S3. La console Amazon S3 s'ouvre.
9. Choisissez Télécharger pour enregistrer vos scripts SQL.

Étape 6 : Appliquer le code converti

La conversion de schéma DMS n'applique pas immédiatement le code converti à la base de données cible. Pour mettre à jour la base de données cible, vous pouvez utiliser les scripts SQL que vous avez créés à l'étape précédente. Vous pouvez également utiliser la procédure suivante pour appliquer le code converti à partir de la conversion de schéma DMS.

Pour appliquer le code converti

1. Dans le volet de la base de données cible, choisissez le schéma de base de données converti. Cochez également la case correspondant au nom de ce schéma.
2. Pour Actions, choisissez Appliquer les modifications. La boîte de dialogue Appliquer les modifications s'affiche.
3. Choisissez Appliquer pour confirmer votre choix.

Étape 7 : Nettoyer et résoudre les problèmes

Vous pouvez utiliser Amazon CloudWatch pour consulter ou partager vos journaux de conversion de schéma DMS.

Pour passer en revue les journaux de conversion de schéma DMS

1. Connectez-vous à la CloudWatch console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Choisissez Journaux, Groupes de journaux.

   Le nom de votre groupe de journaux de conversion de schéma DMS commence par `dms-tasks-sct`. Vous pouvez trier les groupes de journaux par heure de création pour rechercher le groupe de journaux de conversion de schéma DMS.

   De plus, le nom de votre groupe de journaux inclut l'Amazon Resource Name (ARN) de votre projet de migration. Vous pouvez voir l'ARN de votre projet sur la page Projets de migration dans la conversion de schéma DMS. Veillez à choisir ARN dans Préférences.
3. Choisissez le nom de votre groupe de journaux, puis choisissez le nom de votre flux de journaux.
4. Pour Actions, choisissez Exporter les résultats pour enregistrer le journal de conversion de schéma DMS.

Une fois que vous avez terminé la conversion de schéma DMS, nettoyez vos ressources.

Pour nettoyer vos ressources de conversion de schéma DMS

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la.
2. Dans le volet de navigation, choisissez Projets de migration.

   a. Sélectionnez **`sc-project`**.

   b. Choisissez Conversion de schéma, puis choisissez Fermer la conversion de schéma.

   c. Choisissez Supprimer et confirmez votre choix.
3. Dans le volet de navigation, choisissez Profils d'instance.

   a. Sélectionnez **`sc-instance`**.

   b. Choisissez Supprimer et confirmez votre choix.
4. Dans le volet de navigation, choisissez Fournisseurs d'identité.

a. Sélectionnez **`sc-source`** et **`sc-target`**.

b. Choisissez Supprimer et confirmez votre choix.

Assurez-vous également de nettoyer les autres AWS ressources que vous avez créées, telles que votre compartiment Amazon S3, les secrets de base de données AWS Secrets Manager, les rôles IAM et le cloud privé virtuel (VPC).

Configuration d'un réseau pour la conversion de schéma DMS

La conversion de schéma DMS crée une instance de conversion de schéma dans un cloud privé virtuel (VPC) basé sur le service Amazon VPC. Lorsque vous créez votre profil d'instance, vous spécifiez le VPC à utiliser. Vous pouvez utiliser votre VPC par défaut pour votre compte et la Région AWS, ou vous pouvez créer un nouveau VPC.

Vous pouvez utiliser différentes configurations réseau pour configurer l'interaction pour vos bases de données source et cible avec la conversion de schéma DMS. Ces configurations dépendent de l'emplacement de votre fournisseur de données source et de vos paramètres réseau. Les rubriques suivantes décrivent les configurations réseau courantes.

Rubriques

- Utilisation d'un VPC unique pour les fournisseurs de données source et cible
- Utilisation de plusieurs VPC pour les fournisseurs de données source et cible
- Utilisation d'AWS Direct Connect ou d'un VPN pour configurer la connexion d'un réseau à un VPC
- Utilisation d'une connexion Internet à un VPC
- Utilisation d'un environnement sans passerelle Internet

Utilisation d'un VPC unique pour les fournisseurs de données source et cible

La configuration réseau la plus simple pour la conversion de schéma DMS est une configuration de VPC unique. Ici, votre fournisseur de données source, votre profil d'instance et le fournisseur de données cible se trouvent tous dans le même VPC. Vous pouvez utiliser cette configuration pour convertir la base de données source sur une instance Amazon EC2.

Pour utiliser cette configuration, assurez-vous que le groupe de sécurité de VPC utilisé par le profil d'instance a accès aux fournisseurs de données. Par exemple, vous pouvez autoriser une plage de routage inter-domaines sans classe (CIDR) de VPC ou l'adresse IP Elastic pour votre passerelle de traduction d'adresses réseau (NAT).

Utilisation de plusieurs VPC pour les fournisseurs de données source et cible

Si vos fournisseurs de données source et cible se trouvent dans des VPC distincts, vous pouvez créer votre profil d'instance dans l'un de ces VPC. Vous pouvez ensuite lier ces deux VPC en utilisant l'appairage de VPC. Vous pouvez utiliser cette configuration pour convertir la base de données source sur une instance Amazon EC2.

Une connexion d'appairage de VPC est une connexion réseau entre deux VPC qui permet le routage en utilisant les adresses IP privées de chaque VPC comme s'ils figuraient dans le même réseau. Vous pouvez créer une connexion d'appairage de VPC entre vos propres VPC, avec un VPC situé dans un autre compte AWS ou avec un VPC au sein d'une autre Région AWS. Pour de plus amples informations sur l'appairage de VPC, veuillez consulter la section Appairage de VPC dans le Guide de l'utilisateur Amazon VPC.

Pour implémenter l'appairage de VPC, suivez les instructions fournies dans Utilisation de connexions d'appairage de VPC dans le Guide de l'utilisateur Amazon VPC. Assurez-vous que la table de routage d'un VPC contient le bloc CIDR de l'autre. Supposons, par exemple, que le VPC A utilise la destination 10.0.0.0/16 et que le VPC B utilise la destination 172.31.0.0. Dans ce cas, la table de routage du VPC A doit contenir 172.31.0.0, et la table de routage du VPC B doit contenir 10.0.0.0/16. Pour en savoir plus, consultez Mise à jour de vos tables de routage pour une connexion d'appairage de VPC dans la documentation Amazon VPC – Appairage de VPC.

Utilisation d'AWS Direct Connect ou d'un VPN pour configurer la connexion d'un réseau à un VPC

Les réseaux distants peuvent se connecter à un VPC grâce à plusieurs options telles qu'AWS Direct Connect ou une connexion VPN logicielle ou matérielle. Vous pouvez utiliser ces options pour intégrer les services sur site existants en étendant un réseau interne dans le AWS Cloud. Vous pouvez intégrer des services sur site tels que des services de surveillance, d'authentification, de sécurité ou de données, ou encore d'autres systèmes. Grâce à ce type d'extension de réseau, vous pouvez connecter en toute transparence les services sur site à des ressources hébergées par AWS,

telles qu'un VPC. Vous pouvez utiliser cette configuration pour convertir la base de données source sur site.

Dans cette configuration, le groupe de sécurité de VPC doit inclure une règle de routage qui envoie à un hôte le trafic destiné à la plage CIDR d'un VPC ou à une adresse IP spécifique. Cet hôte doit être en mesure d'acheminer le trafic du VPC vers le VPN sur site. Dans ce cas, l'hôte NAT inclut ses propres paramètres de groupe de sécurité. Ces paramètres doivent autoriser le trafic provenant de votre plage CIDR ou de votre groupe de sécurité de VPC à entrer dans l'instance NAT. Pour plus d'informations, consultez Création d'une connexion VPN de site à site dans le Guide de l'utilisateur AWS Site-to-Site VPN.

Utilisation d'une connexion Internet à un VPC

Si vous n'utilisez pas de VPN ni AWS Direct Connect pour vous connecter aux ressources AWS, vous pouvez utiliser une connexion Internet. Cette configuration implique un sous-réseau privé dans un VPC avec une passerelle Internet. La passerelle contient le fournisseur de données cible et le profil d'instance. Vous pouvez utiliser cette configuration pour convertir la base de données source sur site.

Pour ajouter une passerelle Internet à votre VPC, veuillez consulter Attachement d'une passerelle Internet dans le Guide de l'utilisateur Amazon VPC.

La table de routage du VPC doit inclure des règles de routage qui envoient par défaut le trafic non destiné au VPC vers la passerelle Internet. Dans cette configuration, la connexion au fournisseur de données semble provenir de l'adresse IP publique de votre passerelle NAT. Pour plus d'informations, consultez Tables de routage de VPC dans le Guide de l'utilisateur Amazon VPC.

Utilisation d'un environnement sans passerelle Internet

Pour créer un environnement de conversion de schéma sans utiliser de passerelle Internet, procédez comme suit.

1. Suivez les étapes 1 à 3 du didacticiel Premiers pas, avec les modifications suivantes :
   - Choisissez des sous-réseaux privés à la place de sous-réseaux publics.
   - Lors de la création de l'instance, pour Assigner une adresse IP publique, choisissez Non.
2. Ouvrez la console Amazon VPC.
3. Choisissez Points de terminaison, puis Créer un point de terminaison.

4. Sur la page Créer un point de terminaison, procédez comme suit :
   - Pour Catégorie de service, choisissez Services AWS.
   - Dans la liste Services, choisissez com.amazonaws.***{region}***.secretsmanager
   - Dans la section VPC, choisissez le VPC que vous avez créé.
   - Choisissez les sous-réseaux pour votre VPC.
   - Choisissez le groupe de sécurité pour votre VPC.
   - Pour Stratégie, laissez l'option Accès complet sélectionnée.
5. Terminez le reste du didacticiel Premiers pas.

Création de fournisseurs de données sources dans la conversion de schéma DMS

Vous pouvez utiliser une base de données Microsoft SQL Server, Oracle ou PostgreSQL comme fournisseur de données source dans les projets de migration pour la conversion de schémas DMS. Votre fournisseur de données source peut être un moteur autogéré s'exécutant sur site ou sur une instance Amazon EC2.

Veillez à configurer le réseau de manière à autoriser l'interaction entre votre fournisseur de données source et la conversion de schéma DMS. Pour plus d'informations, consultez Configuration d'un réseau pour la conversion de schéma DMS.

Rubriques

- Utilisation d'une base de données Microsoft SQL Server en tant que source dans la conversion de schéma DMS
- Utilisation d'une base de données Oracle en tant que source dans la conversion de schéma DMS
- Utilisation d'une base de données Oracle Data Warehouse en tant que source pour la conversion de schéma DMS
- Utilisation d'une base de données PostgreSQL comme source dans le cadre de la conversion de schéma DMS
- Utilisation d'une base de données MySQL comme source dans la conversion de schéma DMS

Utilisation d'une base de données Microsoft SQL Server en tant que source dans la conversion de schéma DMS

Vous pouvez utiliser les bases de données SQL Server en tant que source de migration dans la conversion de schéma DMS.

Vous pouvez utiliser la conversion de schéma DMS pour convertir des objets de code de base de données de SQL Server vers les cibles suivantes :

- Aurora MySQL
- Aurora PostgreSQL
- RDS for MySQL
- RDS for PostgreSQL

Pour en savoir plus sur les versions de base de données SQL Server prises en charge, consultez Fournisseurs de données sources pour la conversion de schéma DMS.

Pour plus d'informations sur l'utilisation de la conversion de schéma DMS avec une base de données SQL Server source, consultez le guide de migration de SQL Server vers MySQL step-by-step .

Privilèges pour Microsoft SQL Server en tant que source

Voici la liste des privilèges requis pour Microsoft SQL Server en tant que source :

- VIEW DEFINITION
- VIEW DATABASE STATE

Ce privilège `VIEW DEFINITION` permet aux utilisateurs disposant d'un accès public de consulter les définitions des objets. La conversion de schéma DMS utilise le privilège `VIEW DATABASE STATE` pour vérifier les fonctionnalités de l'édition Enterprise de SQL Server.

Répétez l'octroi pour chaque base de données dont vous convertissez le schéma.

En outre, accordez les privilèges suivants sur la base de données `master` :

- VIEW SERVER STATE
- VIEW ANY DEFINITION

La conversion de schéma DMS utilise le privilège `VIEW SERVER STATE` pour collecter les paramètres et la configuration du serveur. Veillez à accorder le privilège `VIEW ANY DEFINITION` permettant de consulter les fournisseurs de données.

Pour lire des informations sur Microsoft Analysis Services, exécutez la commande suivante sur la base de données `master`.

```
EXEC master..sp_addsrvrolemember @loginame = N'<user_name>', @rolename = N'sysadmin'
```

Dans l'exemple précédent, remplacez l'espace réservé *<user_name>* par le nom de l'utilisateur auquel vous avez précédemment accordé les privilèges requis.

Pour lire des informations sur l'agent SQL Server, ajoutez votre utilisateur au AgentUser rôle SQL. Exécutez la commande suivante sur la base de données `msdb`.

```
EXEC sp_addrolemember <SQLAgentRole>, <user_name>;
```

Dans l'exemple précédent, remplacez l'espace réservé *<SQLAgentRole>* par le nom du rôle SQL Server Agent. Ensuite, remplacez l'espace réservé *<user_name>* par le nom de l'utilisateur auquel vous avez précédemment accordé les privilèges requis. Pour plus d'informations, consultez la section Ajouter un utilisateur au AgentUser rôle SQL dans le guide de l'utilisateur Amazon RDS.

Pour détecter l'expédition de journaux, accordez le privilège `SELECT on dbo.log_shipping_primary_databases` sur la base de données `msdb`.

Pour utiliser l'approche de notification de la réplication du langage de définition de données (DDL), accordez le privilège `RECEIVE ON` *<schema_name>*`.`*<queue_name>* sur vos bases de données sources. Dans cet exemple, remplacez l'espace réservé *<schema_name>* par le nom du schéma de la base de données. Remplacez ensuite l'espace réservé *<queue_name>* par le nom d'une table de file d'attente.

Utilisation d'une base de données Oracle en tant que source dans la conversion de schéma DMS

Vous pouvez utiliser les bases de données Oracle en tant que source de migration dans la conversion de schéma DMS.

Pour vous connecter à la base de données Oracle, utilisez l'ID du système Oracle (SID). Pour trouver le SID Oracle, soumettez la requête suivante à votre base de données Oracle :

```
SELECT sys_context('userenv','instance_name') AS SID FROM dual;
```

Vous pouvez utiliser la conversion de schéma DMS pour convertir des objets de code de base de données d'Oracle Database vers les cibles suivantes :

- Aurora MySQL
- Aurora PostgreSQL
- RDS for MySQL
- RDS for PostgreSQL

Pour en savoir plus sur les versions de base de données Oracle prises en charge, consultez Fournisseurs de données sources pour la conversion de schéma DMS.

Pour plus d'informations sur l'utilisation de la conversion de schéma DMS avec une base de données Oracle source, consultez le guide de migration d'Oracle vers step-by-step PostgreSQL.

Privilèges pour Oracle en tant que source

Les privilèges suivants sont requis pour Oracle en tant que source :

- CONNECT
- SELECT_CATALOG_ROLE
- SELECT ANY DICTIONARY
- SELECT ON SYS.ARGUMENT$

Utilisation d'une base de données Oracle Data Warehouse en tant que source pour la conversion de schéma DMS

Vous pouvez utiliser des bases de données Oracle Data Warehouse en tant que source de migration dans l'outil de conversion de schéma DMS pour convertir des objets de code de base de données et le code d'application vers Amazon Redshift.

Pour plus d'informations sur les versions de base de données Oracle prises en charge, consultez Fournisseurs de données sources pour la conversion de schéma DMS. Pour plus d'informations sur l'utilisation de la conversion de schéma DMS avec une base de données Oracle source, consultez le guide de migration d'Oracle vers step-by-step PostgreSQL.

Privilèges requis pour l'utilisation d'Oracle Data Warehouse en tant que source

Les privilèges suivants sont requis pour utiliser Oracle Data Warehouse en tant que source :

- CONNECT
- SELECT_CATALOG_ROLE
- SELECT ANY DICTIONARY

Paramètres de conversion d'Oracle Data Warehouse vers Amazon Redshift

Pour plus d'informations sur les paramètres de conversion de schéma DMS, consultez Spécification des paramètres de conversion de schéma pour les projets de migration.

Les paramètres de conversion d'Oracle Data Warehouse vers Amazon Redshift sont les suivants :

- Ajouter des commentaires dans le code converti pour les actions ayant le niveau de gravité sélectionné ou un niveau supérieur : ce paramètre limite le nombre de commentaires sur les actions dans le code converti. DMS ajoute des commentaires dans le code converti pour les éléments d'action ayant le niveau de gravité sélectionné ou un niveau supérieur.

  Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.
- Nombre maximal de tables pour le cluster Amazon Redshift cible : ce paramètre définit le nombre maximal de tables que DMS peut appliquer à votre cluster Amazon Redshift cible. Amazon Redshift dispose de quotas qui limitent l'utilisation des tables pour différents types de nœud de cluster. Ce paramètre prend en charge les valeurs suivantes :
  - Auto : DMS détermine le nombre de tables à appliquer à votre cluster Amazon Redshift cible en fonction du type de nœud.
  - Définir une valeur : définissez le nombre de tables manuellement.

  DMS convertit toutes vos tables sources, même si le nombre de tables est supérieur à ce que votre cluster Amazon Redshift peut stocker. DMS stocke le code converti dans votre projet et ne l'applique pas à la base de données cible. Si vous atteignez le quota de tables du cluster Amazon Redshift lorsque vous appliquez le code converti, DMS affiche un message d'avertissement. DMS applique également des tables à votre cluster Amazon Redshift cible jusqu'à ce que le nombre de tables atteigne la limite.

Pour plus d'informations sur les quotas de tables d'Amazon Redshift, consultez la section Quotas et limites d'Amazon Redshift.

- Utiliser la vue UNION ALL : ce paramètre vous permet de définir le nombre maximal de tables cibles que DMS peut créer pour une seule table source.

  Amazon Redshift ne prend pas en charge le partitionnement des tables. Pour émuler le partitionnement des tables et accélérer l'exécution des requêtes, DMS peut migrer chaque partition de votre table source vers une table distincte dans Amazon Redshift. Ensuite, DMS crée une vue incluant les données de toutes les tables cibles qu'il crée.

  DMS détermine automatiquement le nombre de partitions de votre table source. Selon le type de partitionnement de votre table source, ce nombre peut dépasser le quota de tables que vous pouvez appliquer à votre cluster Amazon Redshift. Pour éviter d'atteindre ce quota, entrez le nombre maximal de tables cibles que DMS peut créer pour les partitions d'une seule table source. L'option par défaut est 368 tables, soit une partition par jour pendant un an, plus deux tables pour les partitions NO RANGE et UNKNOWN.
- Les éléments de format datetype que vous utilisez dans le code Oracle sont semblables aux chaînes de format datetime dans Amazon Redshift : utilisez ce paramètre pour convertir des fonctions de formatage de type de données telles que `TO_CHAR`, `TO_DATE` et `TO_NUMBER` avec des éléments de format datetime non pris en charge par Amazon Redshift. Par défaut, DMS utilise les fonctions du pack d'extension pour émuler ces éléments non pris en charge dans le code converti.

  Le modèle de format datetime d'Oracle inclut plus d'éléments que les chaînes de format datetime d'Amazon Redshift. Lorsque votre code source inclut uniquement des éléments de format datetime pris en charge par Amazon Redshift, définissez cette valeur pour éviter l'utilisation des fonctions du pack d'extension dans le code converti. Le fait d'éviter les fonctions du pack d'extension accélère l'exécution du code converti.
- Les éléments de format numérique que vous utilisez dans le code Oracle sont semblables aux chaînes de format numérique d'Amazon Redshift : utilisez ce paramètre pour convertir les fonctions de formatage de type numérique non prises en charge par Amazon Redshift. Par défaut, DMS utilise les fonctions du pack d'extension pour émuler ces éléments non pris en charge dans le code converti.

  Le modèle de format numérique d'Oracle inclut plus d'éléments que les chaînes de format numérique d'Amazon Redshift. Lorsque votre code source inclut uniquement des éléments de format numérique pris en charge par Amazon Redshift, définissez cette valeur pour éviter

l'utilisation des fonctions du pack d'extension dans le code converti. Le fait d'éviter les fonctions du pack d'extension accélère l'exécution du code converti.

- Utiliser la fonction NVL pour émuler le comportement des fonctions LEAD et LAG d'Oracle : si votre code source n'utilise pas les valeurs offset par défaut des fonctions `LEAD` et `LAG`, DMS peut émuler ces dernières avec la fonction `NVL`. Par défaut, DMS déclenche une action pour chaque utilisation des fonctions `LEAD` et `LAG`. L'émulation de ces fonctions à l'aide de `NVL` accélère l'exécution du code converti.
- Émuler le comportement des clés primaires et uniques : définissez ce paramètre pour que DMS imite le comportement des contraintes des clés principales et uniques sur le cluster Amazon Redshift cible. Amazon Redshift n'applique pas les contraintes des clés principales et uniques et les utilise uniquement à titre d'information. Si votre code source utilise des contraintes de clé primaire ou unique, définissez ce paramètre pour que DMS imite leur comportement.
- Utiliser un encodage de compression : définissez ce paramètre pour appliquer un encodage de compression aux colonnes des tables Amazon Redshift. DMS attribue automatiquement un encodage de compression à l'aide de l'algorithme Redshift par défaut. Pour plus d'informations sur l'encodage de compression, consultez Encodages de compression dans le Guide du développeur Amazon Redshift Database.

  Amazon Redshift n'applique pas de compression par défaut aux colonnes définies comme clés de tri et de distribution. Pour appliquer une compression à ces colonnes, sélectionnez Utiliser un encodage de compression pour les colonnes KEY. Vous ne pouvez sélectionner cette option que lorsque vous définissez Utiliser un encodage de compression.

Utilisation d'une base de données PostgreSQL comme source dans le cadre de la conversion de schéma DMS

Vous pouvez utiliser les bases de données PostgreSQL comme source de migration dans DMS Schema Conversion.

Vous pouvez utiliser la conversion de schéma DMS pour convertir les objets de code de base de données de la base de données PostgreSQL vers les cibles suivantes :

- MySQL
- Aurora MySQL

Les privilèges requis pour PostgreSQL en tant que source sont les suivants :

- CONNECT ON DATABASE <database_name>
- USAGE ON SCHEMA <database_name>
- SELECT ON ALL TABLES IN SCHEMA <database_name>
- SELECT ON ALL SEQUENCES IN SCHEMA <database_name>

Utilisation d'une base de données MySQL comme source dans la conversion de schéma DMS

Vous pouvez utiliser les bases de données MySQL comme source de migration dans DMS Schema Conversion.

Vous pouvez utiliser la conversion de schéma DMS pour convertir les objets de code de base de données de la base de données MySQL vers les cibles suivantes :

- PostgreSQL
- Aurora PostgreSQL

Les privilèges requis pour MySQL en tant que source sont les suivants :

- `SELECT ON *.*`
- `SHOW VIEW ON *.*`

Paramètres de conversion MySQL vers PostgreSQL

Pour plus d'informations sur les paramètres de conversion de schéma DMS, consultez Spécification des paramètres de conversion de schéma pour les projets de migration.

Les paramètres de conversion de MySQL vers PostgreSQL sont les suivants :

- Commentaires dans le code SQL converti : définissez ce paramètre pour ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure.

  Valeurs valides :

  - Erreurs uniquement
  - Erreurs et avertissements
  - Tous les messages

Création de fournisseurs de données cibles dans la conversion de schéma DMS

Vous pouvez utiliser les bases de données MySQL et PostgreSQL comme fournisseur de données cible dans les projets de migration pour la conversion de schéma DMS. Votre fournisseur de données cible peut être une instance Amazon EC2, Amazon RDS ou Amazon Aurora.

Rubriques

- Utilisation d'une base de données MySQL en tant que cible dans la conversion de schéma DMS
- Utilisation d'une base de données PostgreSQL en tant que cible dans la conversion de schéma DMS
- Utilisation d'un cluster Amazon Redshift en tant que cible pour la conversion du schéma DMS

Utilisation d'une base de données MySQL en tant que cible dans la conversion de schéma DMS

Vous pouvez utiliser les bases de données MySQL en tant que cible de migration dans la conversion de schéma DMS.

Pour en savoir plus sur les bases de données cibles prises en charge, consultez Fournisseurs de données cibles pour la conversion de schéma DMS.

Privilèges pour MySQL en tant que cible

Les privilèges suivants sont requis pour MySQL en tant que cible :

- `CREATE ON *.*`
- `ALTER ON *.*`
- `DROP ON *.*`
- `INDEX ON *.*`
- `REFERENCES ON *.*`
- `SELECT ON *.*`
- `CREATE VIEW ON *.*`
- `SHOW VIEW ON *.*`
- `TRIGGER ON *.*`

- CREATE ROUTINE ON *.*
- ALTER ROUTINE ON *.*
- EXECUTE ON *.*
- CREATE TEMPORARY TABLES ON *.*
- AWS_LAMBDA_ACCESS
- INSERT, UPDATE ON AWS_ORACLE_EXT.*
- INSERT, UPDATE, DELETE ON AWS_ORACLE_EXT_DATA.*
- INSERT, UPDATE ON AWS_SQLSERVER_EXT.*
- INSERT, UPDATE, DELETE ON AWS_SQLSERVER_EXT_DATA.*
- CREATE TEMPORARY TABLES ON AWS_SQLSERVER_EXT_DATA.*

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';
GRANT CREATE ON *.* TO 'user_name';
GRANT ALTER ON *.* TO 'user_name';
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
GRANT EXECUTE ON *.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON *.* TO 'user_name';
GRANT AWS_LAMBDA_ACCESS TO 'user_name';
GRANT INSERT, UPDATE ON AWS_ORACLE_EXT.* TO 'user_name';
GRANT INSERT, UPDATE, DELETE ON AWS_ORACLE_EXT_DATA.* TO 'user_name';
GRANT INSERT, UPDATE ON AWS_SQLSERVER_EXT.* TO 'user_name';
GRANT INSERT, UPDATE, DELETE ON AWS_SQLSERVER_EXT_DATA.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON AWS_SQLSERVER_EXT_DATA.* TO 'user_name';
```

Dans l'exemple précédent, remplacez *user_name* par le nom de votre utilisateur. Remplacez ensuite *your_password* par un mot de passe sécurisé.

Pour utiliser Amazon RDS for MySQL ou Aurora MySQL en tant que cible, définissez le paramètre `lower_case_table_names` sur 1. Cette valeur signifie que le serveur MySQL traite les identifiants des noms d'objets tels que les tables, les index, les déclencheurs et les bases de données sans distinction entre majuscules et minuscules. Si vous avez activé la journalisation binaire dans votre instance cible, définissez le paramètre `log_bin_trust_function_creators` sur 1. Dans ce cas, vous n'avez pas besoin d'utiliser les caractéristiques `DETERMINISTIC`, `READS SQL DATA` ni `NO SQL` pour créer des fonctions stockées. Pour configurer ces paramètres, créez un nouveau groupe de paramètres de base de données ou modifiez un groupe de paramètres de base de données existant.

Utilisation d'une base de données PostgreSQL en tant que cible dans la conversion de schéma DMS

Vous pouvez utiliser les bases de données PostgreSQL en tant que cible de migration dans la conversion de schéma DMS.

Pour en savoir plus sur les bases de données cibles prises en charge, consultez Fournisseurs de données cibles pour la conversion de schéma DMS.

Privilèges pour PostgreSQL en tant que cible

Pour utiliser PostgreSQL en tant que cible, la conversion de schéma DMS nécessite le privilège `CREATE ON DATABASE`. Créez un utilisateur et accordez-lui ce privilège pour chaque base de données que vous souhaitez utiliser dans le projet de migration pour la conversion de schéma DMS.

Pour utiliser Amazon RDS for PostgreSQL en tant que cible, la conversion de schéma DMS nécessite le rôle `rds_superuser`.

Pour utiliser les synonymes publics convertis, modifiez le chemin de recherche par défaut de base de données à l'aide de la commande suivante.

```
ALTER DATABASE <db_name> SET SEARCH_PATH = "$user", public_synonyms, public;
```

Dans cet exemple, remplacez l'espace réservé *<db_name>* par le nom de la base de données.

Dans PostgreSQL, seul le propriétaire du schéma ou un `superuser` peut supprimer un schéma. Le propriétaire peut supprimer un schéma et tous les objets que celui-ci inclut, même si le propriétaire du schéma ne possède pas certains de ces objets.

Lorsque vous utilisez différents utilisateurs pour convertir et appliquer différents schémas à la base de données cible, un message d'erreur peut s'afficher lorsque la conversion de schéma DMS ne parvient pas à supprimer un schéma. Pour éviter ce message d'erreur, utilisez le rôle `superuser`.

Utilisation d'un cluster Amazon Redshift en tant que cible pour la conversion du schéma DMS

Vous pouvez utiliser des bases de données Amazon Redshift en tant que cible de migration dans la conversion de schéma DMS. Pour en savoir plus sur les bases de données cibles prises en charge, consultez Fournisseurs de données cibles pour la conversion de schéma DMS.

Privilèges d'utilisation d'Amazon Redshift en tant que cible

L'utilisation d'Amazon Redshift en tant que cible pour la conversion du schéma DMS nécessite les privilèges suivants :

- CREATE ON DATABASE : permet à DMS de créer des schémas dans la base de données.
- CREATE ON SCHEMA : permet à DMS de créer des objets dans le schéma de la base de données.
- GRANT USAGE ON LANGUAGE : permet à DMS de créer des fonctions et procédures dans la base de données.
- GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog : fournit à l'utilisateur des informations système sur le cluster Amazon Redshift.
- GRANT SELECT ON pg_class_info : fournit à l'utilisateur des informations sur le style de distribution des tables.

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et lui accorder des autorisations. Remplacez les exemples de valeurs par vos propres valeurs.

```
CREATE USER user_name PASSWORD your_password;
GRANT CREATE ON DATABASE db_name TO user_name;
GRANT CREATE ON SCHEMA schema_name TO user_name;
GRANT USAGE ON LANGUAGE plpythonu TO user_name;
GRANT USAGE ON LANGUAGE plpgsql TO user_name;
GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog TO user_name;
GRANT SELECT ON pg_class_info TO user_name;
GRANT SELECT ON sys_serverless_usage TO user_name;
GRANT SELECT ON pg_database_info TO user_name;
```

```
GRANT SELECT ON pg_statistic TO user_name;
```

Répétez l'opération `GRANT CREATE ON SCHEMA` pour chaque schéma cible dans lequel vous allez appliquer le code converti ou migrer des données.

Vous pouvez appliquer un pack d'extension à votre base de données Amazon Redshift cible. Le pack d'extension est un module complémentaire qui émule les fonctions de la base de données source requises pour convertir des objets vers la base de données cible. Pour de plus amples informations, veuillez consulter Utilisation de packs d'extension dans la conversion de schéma DMS.

Gestion de projets de migration pour la conversion de schéma DMS

Après avoir créé un profil d'instance et des fournisseurs de données compatibles pour la conversion de schéma, créez un projet de migration. Pour de plus amples informations, veuillez consulter Création de projets de migration.

Pour utiliser ce nouveau projet dans la conversion de schéma DMS, sur la page Projets de migration, choisissez votre projet dans la liste. Ensuite, dans l'onglet Conversion de schéma, choisissez Lancer la conversion de schéma.

Le premier lancement de la conversion de schéma DMS nécessite une certaine configuration. AWS Database Migration Service (AWS DMS) démarre une instance de conversion de schéma, ce qui peut prendre jusqu'à 15 minutes. Ce processus lit également les métadonnées des bases de données source et cible. Après un premier lancement réussi, vous pouvez accéder plus rapidement à la conversion de schéma DMS.

Amazon résilie l'instance de conversion de schéma utilisée par votre projet de migration dans les trois jours suivant la fin du projet. Vous pouvez récupérer votre schéma converti et votre rapport d'évaluation à partir du compartiment Amazon S3 que vous utilisez pour la conversion de schéma DMS.

Spécification des paramètres de projet de migration pour la conversion de schéma DMS

Après avoir créé votre projet de migration et lancé la conversion de schéma, vous pouvez spécifier les paramètres du projet de migration. Vous pouvez modifier les paramètres de conversion pour améliorer les performances du code converti. Vous pouvez également personnaliser la vue de conversion de schéma.

Les paramètres de conversion dépendent de vos plateformes de base de données source et cible. Pour plus d'informations, consultez Création de fournisseurs de données sources et Création de fournisseurs de données cibles.

Pour spécifier les schémas et les bases de données que vous souhaitez voir apparaître dans les volets des bases de données source et cible, utilisez les paramètres de vue arborescente. Vous pouvez masquer les schémas vides, les bases de données vides, les bases de données système et les bases de données ou schémas définis par l'utilisateur.

Pour masquer les bases de données et les schémas dans la vue d'arborescence

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS DMS à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration, puis dans l'onglet Conversion de schéma, choisissez Lancer la conversion de schéma.
4. Sélectionnez Paramètres. La page Paramètres s'ouvre.
5. Dans la section Arborescence, effectuez les opérations suivantes :

   - Choisissez Masquer les schémas vides pour masquer les schémas vides.
   - Choisissez Masquer les bases de données vides pour masquer les bases de données vides.
   - Pour Bases de données ou schémas système, choisissez les bases de données et les schémas système par leur nom pour les masquer.
   - Pour Bases de données ou schémas définis par l'utilisateur, entrez les noms des schémas et bases de données définis par l'utilisateur que vous voulez masquer. Choisissez Ajouter. Ces noms ne sont pas sensibles à la casse.

     Pour ajouter plusieurs bases de données ou schémas, séparez leurs noms par une virgule. Pour ajouter plusieurs objets portant des noms similaires, utilisez le symbole de pourcentage (%) comme caractère générique. Ce caractère générique remplace un nombre quelconque de symboles quelconques dans le nom de la base de données ou du schéma.

   Répétez ces étapes pour les sections Source et Cible.
6. Choisissez Appliquer, puis Conversion de schéma.

Création de rapports d'évaluation de migration de base de données avec la conversion de schéma DMS

Le rapport que génère la conversion de schéma DMS pour vous aider à convertir votre schéma constitue un élément important de cette fonctionnalité. Ce rapport d'évaluation de migration de base de données résume toutes les tâches de conversion de schéma. Il détaille également les éléments d'action du schéma qui ne peuvent pas être convertis vers le moteur de base de données de votre instance de base de données cible. Vous pouvez afficher le rapport dans la console AWS DMS ou enregistrer une copie de ce rapport au format PDF ou CSV.

Le rapport d'évaluation de migration comprend les éléments suivants :

- Une synthèse
- Des recommandations, y compris la conversion d'objets serveur, des suggestions de sauvegarde et les modifications de serveur liées

Lorsque la conversion de schéma DMS ne peut pas convertir automatiquement certains de vos éléments, le rapport fournit des estimations des efforts à fournir pour écrire le code équivalent pour votre instance de base de données cible.

Rubriques

- Création d'un rapport d'évaluation de migration de base de données
- Affichage de votre rapport d'évaluation de migration de base de données
- Enregistrement de votre rapport d'évaluation de migration de base de données

Création d'un rapport d'évaluation de migration de base de données

Après avoir créé un projet de migration, suivez la procédure ci-dessous pour créer un rapport d'évaluation de migration de base de données.

Pour créer un rapport d'évaluation de migration de base de données

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS DMS à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration, puis choisissez Conversion de schéma.

4. Choisissez Lancer la conversion de schéma. La page Conversion de schéma s'ouvre.
5. Dans le volet de la base de données source, choisissez le schéma de base de données ou les éléments de schéma que vous souhaitez évaluer. Pour inclure plusieurs objets dans le rapport, assurez-vous de sélectionner tous les éléments.
6. Après avoir coché les cases correspondant à tous les objets de schéma que vous souhaitez évaluer, vous devez choisir le nœud parent pour les objets sélectionnés. Le menu Actions du volet de la base de données source est désormais disponible.
7. Choisissez Évaluer dans le menu Actions. Une boîte de dialogue de confirmation s'affiche.
8. Choisissez Évaluer dans cette boîte de dialogue pour confirmer votre choix.

Affichage de votre rapport d'évaluation de migration de base de données

Après avoir créé un rapport d'évaluation, la conversion de schéma DMS ajoute des informations dans les onglets suivants :

- Récapitulatif
- Éléments d'actions

L'onglet Résumé indique le nombre d'éléments que la conversion de schéma DMS peut convertir automatiquement.

L'onglet Éléments d'actions affiche les éléments que la conversion de schéma DMS ne peut pas convertir automatiquement et fournit des recommandations sur la manière de gérer ces éléments.

Récapitulatif du rapport d'évaluation

L'onglet Summary affiche un récapitulatif du rapport d'évaluation de migration de base de données. Il indique le nombre d'éléments que la conversion de schéma DMS peut convertir automatiquement pour les objets de stockage de base de données et les objets de code de base de données.

Dans la plupart des cas, la conversion de schéma DMS ne peut pas convertir automatiquement tous les éléments de schéma vers le moteur de base de données cible. L'onglet Résumé fournit une estimation des efforts à fournir pour créer les mêmes éléments de schéma que ceux de votre source dans votre instance de base de données cible.

Pour consulter le résumé de conversion pour les objets de stockage de base de données tels que les tables, les séquences, les contraintes, les types de données, etc., choisissez Objets de stockage de base de données.

Pour consulter le résumé de conversion pour les objets de code de base de données tels que les procédures, les fonctions, les vues, les déclencheurs, etc., choisissez Objets de code de base de données.

Pour modifier la portée du rapport d'évaluation, sélectionnez le nœud requis dans l'arborescence de la base de données source. La conversion de schéma DMS met à jour le résumé du rapport d'évaluation pour qu'il corresponde à la portée sélectionnée.

Éléments d'action du rapport d'évaluation

L'onglet Éléments d'actions contient une liste d'éléments que la conversion de schéma DMS ne peut pas convertir automatiquement dans un format compatible avec le moteur de base de données cible. Pour chaque élément d'action, la conversion de schéma DMS fournit la description du problème et l'action recommandée. La conversion de schéma DMS regroupe les éléments d'action similaires et affiche le nombre d'occurrences.

Pour afficher le code de l'objet de base de données associé, sélectionnez un élément d'action dans la liste.

Enregistrement de votre rapport d'évaluation de migration de base de données

Après avoir créé un rapport d'évaluation de migration de base de données, vous pouvez enregistrer une copie de ce rapport au format PDF ou CSV.

Pour enregistrer un rapport d'évaluation de migration de base de données au format PDF

1. Choisissez Exporter, puis PDF. Passez en revue la boîte de dialogue, puis choisissez Exporter au format PDF.
2. La conversion de schéma DMS crée une archive avec votre fichier PDF et stocke cette archive dans votre compartiment Amazon S3. Pour changer de compartiment Amazon S3, modifiez les paramètres de conversion de schéma dans votre profil d'instance.
3. Ouvrez le fichier du rapport d'évaluation dans votre compartiment Amazon S3.

Pour enregistrer un rapport d'évaluation de migration de base de données au format CSV

1. Choisissez Exporter, puis CSV. Passez en revue la boîte de dialogue, puis choisissez Exporter au format CSV.
2. La conversion de schéma DMS crée une archive avec les fichiers CSV et stocke cette archive dans votre compartiment Amazon S3. Pour changer de compartiment Amazon S3, modifiez les paramètres de conversion de schéma dans votre profil d'instance.
3. Ouvrez les fichiers de rapport d'évaluation dans votre compartiment Amazon S3.

Le fichier PDF contient à la fois le récapitulatif et les informations sur les éléments d'action.

Lorsque vous exportez votre rapport d'évaluation au format CSV, la conversion de schéma DMS crée trois fichiers CSV.

Le premier fichier CSV contient les informations suivantes sur les éléments d'action :

- Catégorie
- Occurrence
- Elément d'action
- Sujet
- Groupe
- Description
- Références de documentation
- Action recommandée
- Line
- Position
- Source
- Cible
- Adresse IP et port du serveur
- Database (Base de données)
- Schema (Schéma)

Le deuxième fichier CSV inclut le suffixe `Action_Items_Summary` dans son nom et contient les informations suivantes :

- Schema
- Elément d'action
- Nombre d'occurrences
- Efforts liés à la courbe d'apprentissage, c'est-à-dire la quantité d'efforts à fournir pour concevoir une approche permettant de convertir chaque élément d'action
- Efforts pour convertir une occurrence de l'élément d'action, qui montre l'effort à fournir pour convertir chaque élément d'action, conformément à l'approche conçue
- Description de l'élément d'action
- Action recommandée

Les valeurs qui indiquent le niveau d'efforts à fournir sont basées sur une échelle pondérée, allant de faible (niveau minimal) à élevé (niveau maximal).

Le troisième fichier CSV inclut `Summary` dans son nom et contient les informations suivantes :

- Catégorie
- Nombre d'objets
- Objets automatiquement convertis
- Objets avec des actions simples
- Objets avec des actions de complexité moyenne
- Objets avec des actions complexes
- Nombre total de lignes de code

Utilisation de la conversion de schéma DMS

La conversion de schéma DMS convertit vos schémas de base de données existants et la majeure partie des objets de code de base de données dans un format compatible avec la base de données cible.

La conversion de schéma DMS automatise une grande partie du processus de conversion de vos schémas de base de données de traitement des transactions en ligne (OLTP) au format Amazon RDS for MySQL ou RDS for PostgreSQL. Les moteurs de base de données source et cible contiennent de nombreuses fonctionnalités et capacités, et la conversion de schéma DMS tente de créer un schéma équivalent dans la mesure du possible. Pour les objets de base de données pour

lesquels la conversion directe n'est pas possible, la conversion de schéma DMS fournit une liste d'actions à effectuer.

Pour convertir votre schéma de base de données, utilisez le processus suivant :

- Avant de convertir vos schémas de base de données, configurez des règles de transformation qui modifient les noms de vos objets de base de données lors de la conversion.
- Créez un rapport d'évaluation de la migration de base de données pour estimer la complexité de la migration. Ce rapport fournit des détails sur les éléments de schéma que la conversion de schéma DMS ne peut pas convertir automatiquement.
- Convertissez les objets de stockage et de code de la base de données source. La conversion de schéma DMS crée une version locale des objets de base de données convertis. Vous pouvez accéder à ces objets convertis dans votre projet de migration.
- Enregistrez le code converti dans des fichiers SQL pour consulter, modifier ou traiter les éléments d'action de conversion. Vous pouvez éventuellement appliquer le code converti directement à la base de données cible.

Pour convertir les schémas d'entrepôt de données, utilisez le bureau AWS Schema Conversion Tool. Pour plus d'informations, consultez Conversion de schémas d'entrepôt de données vers Amazon Redshift dans le Guide de l'utilisateur de l'Outil de conversion des schémas AWS .

Rubriques

- Configuration des règles de transformation dans la conversion de schéma DMS
- Conversion de schémas de base de données dans la conversion de schéma DMS
- Spécification des paramètres de conversion de schéma pour les projets de migration
- Actualisation de vos schémas de base de données dans la conversion de schéma DMS
- Enregistrement et application de votre code converti dans la conversion de schéma DMS

Configuration des règles de transformation dans la conversion de schéma DMS

Avant de convertir votre schéma de base de données avec la conversion de schéma DMS, vous pouvez configurer des règles de transformation. Les règles de transformation peuvent notamment modifier le nom d'un objet en minuscules ou en majuscules, ajouter ou supprimer un préfixe ou un suffixe et renommer des objets. Supposons par exemple que vous disposiez d'un ensemble de

tables dans votre schéma source nommé `test_TABLE_NAME`. Vous pouvez configurer une règle qui remplace le préfixe `test_` par le préfixe `demo_` dans le schéma cible.

Vous pouvez créer des règles de transformation qui effectuent les tâches suivantes :

- Ajouter, supprimer ou remplacer un préfixe
- Ajouter, supprimer ou remplacer un suffixe
- Modifier le type de données d'une colonne
- Changez le nom d'un objet en minuscules ou en majuscules
- Renommer des objets

Vous pouvez créer des règles de transformation pour les objets suivants :

- Schema
- Tableau
- Colonne

Création de règles de transformation

La conversion de schéma DMS stocke les règles de transformation dans le cadre de votre projet de migration. Vous pouvez configurer des règles de transformation lorsque vous créez votre projet de migration ou les modifier ultérieurement.

Vous pouvez ajouter plusieurs règles de transformation dans votre projet. La conversion de schéma DMS applique les règles de transformation lors de la conversion dans l'ordre dans lequel vous les avez ajoutées.

Pour créer des règles de transformation

1. Sur la page Créer un projet de migration, choisissez Ajouter une règle de transformation. Pour de plus amples informations, veuillez consulter Création de projets de migration.
2. Sous Objectif de la règle, choisissez le type des objets de base de données auxquels cette règle s'applique.
3. Pour Schéma source, choisissez Entrer un schéma. Entrez ensuite les noms de vos schémas, tables et colonnes sources auxquels cette règle s'applique. Vous pouvez entrer un nom exact pour sélectionner un objet, ou spécifier un modèle afin de sélectionner plusieurs objets. Utilisez

le signe du pourcentage (%) comme caractère générique pour remplacer un certain nombre de symboles dans le nom de l'objet de base de données.

4. Pour Action, choisissez la tâche à effectuer.
5. En fonction du type de règle, entrez une ou deux valeurs supplémentaires. Par exemple, pour renommer un objet, entrez le nouveau nom de l'objet. Pour remplacer un préfixe, entrez l'ancien préfixe et le nouveau préfixe.
6. Choisissez Ajouter une règle de transformation pour ajouter une autre règle de transformation.

   Une fois que vous avez terminé d'ajouter des règles, choisissez Créer un projet de migration.

Pour dupliquer une règle de transformation existante, choisissez Dupliquer. Pour modifier une règle de transformation existante, choisissez-la dans la liste. Pour supprimer une règle de transformation existante, choisissez Supprimer.

Modification de règles de transformation

Vous pouvez ajouter de nouvelles règles de transformation, supprimer ou modifier des règles de transformation existantes dans votre projet de migration. Dans la mesure où la conversion de schéma DMS applique les règles de transformation lors du lancement de la conversion de schéma, assurez-vous de fermer la conversion de schéma et de la relancer après avoir modifié vos règles.

Pour modifier des règles de transformation

1. Connectez-vous à la AWS Management Console AWS DMS console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration, puis choisissez votre projet de migration.
3. Choisissez Conversion de schéma, puis choisissez Fermer la conversion de schéma.
4. Une fois AWS DMS la conversion du schéma terminée, choisissez Modifier pour modifier les paramètres de votre projet de migration.
5. Pour Règles de transformation, choisissez l'une des actions suivantes :

   - Choisissez Dupliquer pour dupliquer une règle de transformation existante et l'ajouter à la fin de la liste.
   - Choisissez Supprimer pour supprimer une règle de transformation existante.
   - Choisissez la règle de transformation existante pour la modifier.
6. Une fois que vous avez terminé de modifier les règles, choisissez Enregistrer les modifications.

7. Sur la page Projets de migration, choisissez votre projet dans la liste. Choisissez Conversion de schéma, puis Lancer la conversion de schéma.

Conversion de schémas de base de données dans la conversion de schéma DMS

Après avoir créé le projet de migration et l'avoir connecté à vos bases de données sources et cibles, vous pouvez convertir vos objets de base de données source dans un format compatible avec la base de données cible. La conversion de schéma DMS affiche votre schéma de base de données source dans le panneau de gauche sous forme d'arborescence.

Chaque nœud de l'arborescence de base de données est avec chargement différé. Lorsque vous choisissez un nœud dans l'arborescence, la conversion de schéma DMS demande les informations de schéma de la base de données source à ce moment-là. Pour charger les informations du schéma plus rapidement, choisissez votre schéma, puis choisissez Charger les métadonnées dans le menu Actions. La conversion de schéma DMS lit ensuite les métadonnées de la base de données et stocke les informations dans un compartiment Amazon S3. Vous pouvez désormais parcourir les objets de base de données plus rapidement.

Vous pouvez convertir l'ensemble du schéma de base de données ou choisir n'importe quel élément de schéma à convertir dans la base de données source. Si l'élément de schéma que vous choisissez dépend d'un élément parent, la conversion de schéma DMS génère également le schéma de l'élément parent. Par exemple, lorsque vous choisissez une table à convertir, la conversion de schéma DMS crée la table convertie et le schéma de base de données dans lequel se trouve la table.

Conversion d'objets de base de données

Vous pouvez utiliser la conversion de schéma DMS pour convertir un schéma de base de données entier ou des objets de schéma de base de données séparés.

Pour convertir un schéma de base de données entier

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration, puis choisissez Conversion de schéma.
4. Choisissez Lancer la conversion de schéma. La page Conversion de schéma s'ouvre.
5. Dans le volet de la base de données source, cochez la case en regard du nom de schéma.

6. Choisissez ce schéma dans le volet de gauche du projet de migration. La conversion de schéma DMS met en évidence le nom du schéma en bleu et active le menu Actions.
7. Pour Actions, choisissez Convertir. La boîte de dialogue de conversion s'affiche.
8. Choisissez Convertir dans cette boîte de dialogue pour confirmer votre choix.

Pour convertir vos objets de base de données source

1. Connectez-vous à la AWS Management Console AWS DMS console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration, puis choisissez Conversion de schéma.
4. Choisissez Lancer la conversion de schéma. La page Conversion de schéma s'ouvre.
5. Dans le volet de la base de données source, sélectionnez vos objets de base de données source.
6. Après avoir coché toutes les cases correspondant aux objets que vous souhaitez convertir, choisissez le nœud parent pour tous les objets sélectionnés dans le volet de gauche.

   La conversion de schéma DMS met en évidence le nœud parent en bleu et active le menu Actions.
7. Pour Actions, choisissez Convertir. La boîte de dialogue de conversion s'affiche.
8. Choisissez Convertir dans cette boîte de dialogue pour confirmer votre choix.

Par exemple, pour convertir deux tables sur dix, cochez les cases correspondant aux deux tables que vous souhaitez convertir. Notez que le menu Actions est inactif. Après avoir choisi le nœud Tables, la conversion de schéma DMS met en évidence son nom en bleu et active le menu Actions. Vous pouvez ensuite choisir Convertir dans ce menu.

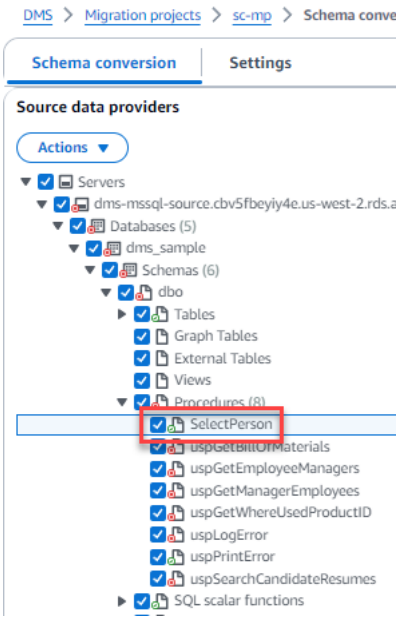
De même, pour convertir deux tables et trois procédures, cochez les cases correspondant aux noms des objets. Choisissez ensuite le nœud de schéma pour activer le menu Actions, puis choisissez Convertir le schéma.

Modification et enregistrement de votre code SQL converti

La page de conversion du schéma vous permet de modifier le code SQL converti dans les objets de votre base de données. Utilisez la procédure suivante pour modifier votre code SQL converti, appliquer les modifications et les enregistrer.

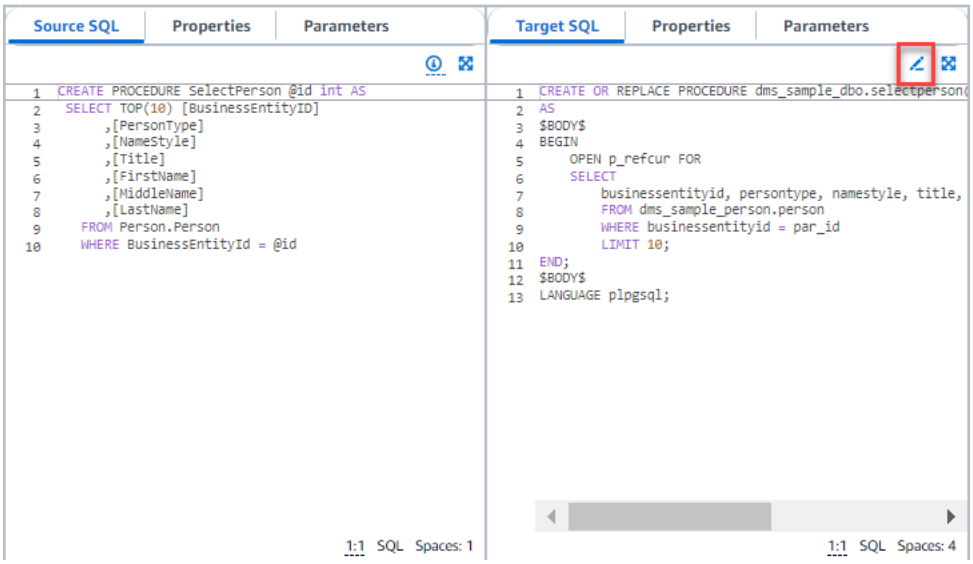
Pour modifier, appliquer des modifications à votre code converti de SQL et l'enregistrer

1. Sur la page de conversion du schéma, ouvrez l'arborescence dans le volet Fournisseurs de données source pour afficher un objet de code.



2. Dans le volet Fournisseurs de données source, choisissez Actions, Convertir. Confirmez l'action.

3. Lorsque la conversion est terminée, pour afficher le code SQL converti, agrandissez le volet central si nécessaire. Pour modifier le code SQL converti, cliquez sur l'icône de modification dans le volet Target SQL.



4. Après avoir modifié le code SQL cible, confirmez vos modifications en cliquant sur l'icône de vérification en haut de la page. Confirmez l'action.

5. Dans le volet Fournisseurs de données cibles, choisissez Actions, Appliquer les modifications. Confirmez l'action.
6. DMS écrit la procédure modifiée dans le magasin de données cible.

Examen d'objets de base de données convertis

Après avoir converti vos objets de base de données source, vous pouvez choisir un objet dans le volet de gauche de votre projet. Vous pouvez ensuite afficher le code source et le code converti de cet objet. La conversion de schéma DMS charge automatiquement le code converti pour l'objet que vous avez sélectionné dans le volet de gauche. Vous pouvez également consulter les propriétés ou les paramètres de l'objet que vous avez sélectionné.

La conversion de schéma DMS stocke automatiquement le code converti dans le cadre de votre projet de migration. Elle n'applique pas ces modifications de code à la base de données cible. Pour plus d'informations sur l'application du code converti à la base de données cible, consultez Application de votre code converti. Pour supprimer le code converti de votre projet de migration, sélectionnez votre schéma cible dans le volet de droite, puis choisissez Actualiser à partir de la base de données dans le menu Actions.

Après avoir converti vos objets de base de données source, vous pouvez voir le résumé de la conversion et les éléments d'action dans le volet central inférieur. Vous pouvez voir les mêmes informations lorsque vous créez un rapport d'évaluation. Le rapport d'évaluation est utile pour identifier et résoudre les éléments de schéma que la conversion de schéma DMS ne peut pas convertir. Vous pouvez enregistrer le résumé du rapport d'évaluation et la liste des éléments d'action de conversion dans des fichiers CSV. Pour de plus amples informations, veuillez consulter Rapports d'évaluation de migration de base de données.

Spécification des paramètres de conversion de schéma pour les projets de migration

Après avoir créé un projet de migration, vous pouvez spécifier les paramètres de conversion dans la conversion de schéma DMS. La configuration de vos paramètres de conversion de schéma permet d'améliorer les performances du code converti.

Pour modifier les paramètres de conversion

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.

2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration. Choisissez Conversion de schéma, puis Lancer la conversion de schéma.
4. Sélectionnez Settings (Paramètres). La page Paramètres s'ouvre.
5. Dans la section Conversion, modifiez les paramètres.
6. Choisissez Appliquer, puis Conversion de schéma.

Pour toutes les paires de conversion, vous pouvez limiter le nombre de commentaires contenant des éléments d'action dans le code converti. Pour limiter le nombre de commentaires dans le code converti, ouvrez les paramètres de conversion dans votre projet de migration.

Pour Commentaires dans le code SQL converti, choisissez le niveau de gravité des éléments d'action. La conversion de schéma DMS ajoute des commentaires dans le code converti pour les éléments d'action de gravité sélectionnée et supérieure. Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement.

Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

Les autres paramètres de conversion sont différents pour chaque paire de bases de données sources et cibles.

Rubriques

- Paramètres de conversion Oracle vers MySQL
- Paramètres de conversion Oracle vers PostgreSQL
- Paramètres de conversion SQL Server vers MySQL
- Paramètres de conversion SQL Server vers PostgreSQL
- Paramètres de conversion de PostgreSQL vers MySQL

Paramètres de conversion Oracle vers MySQL

Les paramètres de conversion Oracle vers MySQL dans la conversion de schéma DMS sont les suivants :

- Votre base de données Oracle source peut utiliser la pseudocolonne `ROWID`. MySQL ne prend pas en charge de fonctionnalité similaire. La conversion de schéma DMS peut émuler la pseudocolonne `ROWID` dans le code converti. Pour ce faire, activez l'option Générer l'ID de ligne.

  Si votre code Oracle source n'utilise pas la pseudocolonne `ROWID`, désactivez l'option Générer l'ID de ligne. Dans ce cas, le code converti fonctionne plus rapidement.

- Votre code Oracle source peut inclure les fonctions `TO_CHAR`, `TO_DATE` et `TO_NUMBER` avec des paramètres que MySQL ne prend pas en charge. Par défaut, la conversion de schéma DMS émule l'utilisation de ces paramètres dans le code converti.

  Vous pouvez utiliser les fonctions `TO_CHAR`, `TO_DATE` et `TO_NUMBER` MySQL natives lorsque votre code Oracle source ne contient pas de paramètres non pris en charge par MySQL. Dans ce cas, le code converti fonctionne plus rapidement. Pour ce faire, sélectionnez les valeurs suivantes :

  - Utiliser une fonction MySQL TO_CHAR native
  - Utiliser une fonction MySQL TO_DATE native
  - Utiliser une fonction MySQL TO_NUMBER native

- Votre base de données et vos applications peuvent s'exécuter dans différents fuseaux horaires. Par défaut, la conversion de schéma DMS émule les fuseaux horaires dans le code converti. Toutefois, vous n'avez pas besoin de cette émulation lorsque votre base de données et vos applications utilisent le même fuseau horaire. Dans ce cas, sélectionnez Améliorer les performances du code converti lorsque la base de données et les applications utilisent le même fuseau horaire.

Paramètres de conversion Oracle vers PostgreSQL

Les paramètres de conversion Oracle vers PostgreSQL dans la conversion de schéma DMS sont les suivants :

- AWS DMS peut convertir les vues matérialisées Oracle en tables ou en vues matérialisées sur PostgreSQL. Pour Vues matérialisées, choisissez le mode de conversion de vos vues matérialisées sources.
- Votre base de données Oracle source peut utiliser la pseudocolonne `ROWID`. PostgreSQL ne prend pas en charge de fonctionnalité similaire. La conversion de schéma DMS peut émuler la pseudocolonne `ROWID` dans le code converti à l'aide du type de données `bigint` ou `character varying`. Pour ce faire, choisissez Utiliser le type de données bigint pour émuler

la pseudocolonne ROWID ou Utiliser le type de données character varying pour émuler la pseudocolonne ROWID pour ID de ligne.

Si votre code Oracle source n'utilise pas la pseudocolonne `ROWID`, choisissez Ne pas générer. Dans ce cas, le code converti fonctionne plus rapidement.

- Votre code Oracle source peut inclure les fonctions `TO_CHAR`, `TO_DATE` et `TO_NUMBER` avec des paramètres que PostgreSQL ne prend pas en charge. Par défaut, la conversion de schéma DMS émule l'utilisation de ces paramètres dans le code converti.

  Vous pouvez utiliser les fonctions `TO_CHAR`, `TO_DATE` et `TO_NUMBER` PostgreSQL natives lorsque votre code Oracle source ne contient pas de paramètres non pris en charge par PostgreSQL. Dans ce cas, le code converti fonctionne plus rapidement. Pour ce faire, sélectionnez les valeurs suivantes :

  - Utiliser une fonction PostgreSQL TO_CHAR native
  - Utiliser une fonction PostgreSQL TO_DATE native
  - Utiliser une fonction PostgreSQL TO_NUMBER native
- Votre base de données et vos applications peuvent s'exécuter dans différents fuseaux horaires. Par défaut, la conversion de schéma DMS émule les fuseaux horaires dans le code converti. Toutefois, vous n'avez pas besoin de cette émulation lorsque votre base de données et vos applications utilisent le même fuseau horaire. Dans ce cas, sélectionnez Améliorer les performances du code converti lorsque la base de données et les applications utilisent le même fuseau horaire.
- Pour continuer à utiliser des séquences dans votre code converti, sélectionnez Remplir les séquences converties avec la dernière valeur générée côté source.
- Dans certains cas, la base de données Oracle source peut uniquement stocker des valeurs entières dans les colonnes de clé primaire ou étrangère du type de données `NUMBER`. Dans ces cas, AWS DMS vous pouvez convertir ces colonnes en type de `BIGINT` données. Cette approche améliore les performances de votre code converti. Pour ce faire, sélectionnez Convertir les colonnes de clé primaire et étrangère du type de données NUMBER en type de données BIGINT. Assurez-vous que votre source n'inclut pas de valeurs à virgule flottante dans ces colonnes pour éviter toute perte de données.
- Pour ignorer les déclencheurs et les contraintes désactivés dans votre code source, choisissez Convertir uniquement les déclencheurs et les contraintes actifs.
- Vous pouvez utiliser la conversion de schéma DMS pour convertir des variables de chaîne appelées en tant que code SQL dynamique. Le code de la base de données peut modifier les

valeurs de ces variables de chaîne. Pour vous assurer que cela convertit AWS DMS toujours la dernière valeur de cette variable de chaîne, sélectionnez Convertir le code SQL dynamique créé dans des routines appelées.

- PostgreSQL versions 10 et antérieures ne prend pas en charge les procédures. Si vous n'êtes pas habitué à utiliser les procédures dans PostgreSQL AWS DMS , vous pouvez convertir les procédures Oracle en fonctions PostgreSQL. Pour ce faire, sélectionnez Convertir les procédures en fonctions.
- Pour obtenir des informations supplémentaires sur les éléments d'action qui se sont produits, vous pouvez ajouter des fonctions spécifiques au pack d'extension. Pour ce faire, sélectionnez Ajouter des fonctions de pack d'extension qui déclenchent des exceptions définies par l'utilisateur. Choisissez ensuite les niveaux de gravité pour déclencher les exceptions définies par l'utilisateur. Assurez-vous d'appliquer le schéma du pack d'extension après avoir converti vos objets de base de données source. Pour plus d'informations sur les packs d'extension, consultez Utilisation de packs d'extension.
- Votre base de données Oracle source peut inclure des contraintes avec les noms générés automatiquement. Si votre code source utilise ces noms, assurez-vous de sélectionner Conserver les noms des contraintes générées par le système. Si votre code source utilise ces contraintes, mais pas leur nom, désactivez cette option pour augmenter la vitesse de conversion.
- Si vos bases de données sources et cibles s'exécutent dans des fuseaux horaires différents, la fonction qui émule la fonction Oracle intégrée `SYSDATE` renvoie des valeurs différentes de celles de la fonction source. Pour garantir que vos fonctions source et cible renvoient les mêmes valeurs, choisissez Définir le fuseau horaire de la base de données source.
- Vous pouvez utiliser les fonctions de l'extension orafce dans votre code converti. Pour ce faire, pour Routines intégrées orafce, sélectionnez les fonctions à utiliser. Pour plus d'informations sur Oracle, voir orace on. GitHub

Paramètres de conversion SQL Server vers MySQL

Les paramètres de conversion SQL Server vers MySQL dans la conversion de schéma DMS sont les suivants :

- Votre base de données SQL Server source peut stocker le résultat de `EXEC` dans une table. La conversion de schéma DMS crée des tables temporaires et une procédure supplémentaire pour émuler cette fonctionnalité. Pour utiliser cette émulation, sélectionnez Créer des routines supplémentaires pour gérer les jeux de données ouverts.

Paramètres de conversion SQL Server vers PostgreSQL

Les paramètres de conversion SQL Server vers PostgreSQL dans la conversion de schéma DMS sont les suivants :

- Dans SQL Server, vous pouvez utiliser des index portant le même nom dans différentes tables. Toutefois, dans PostgreSQL, tous les noms d'index que vous utilisez dans le schéma doivent être uniques. Pour vous assurer que la conversion de schéma DMS génère des noms uniques pour tous vos index, sélectionnez Générer des noms uniques pour les index.
- PostgreSQL versions 10 et antérieures ne prend pas en charge les procédures. Si vous n'êtes pas habitué à utiliser les procédures dans PostgreSQL AWS DMS , vous pouvez convertir les procédures SQL Server en fonctions PostgreSQL. Pour ce faire, sélectionnez Convertir les procédures en fonctions.
- Votre base de données SQL Server source peut stocker le résultat de `EXEC` dans une table. La conversion de schéma DMS crée des tables temporaires et une procédure supplémentaire pour émuler cette fonctionnalité. Pour utiliser cette émulation, sélectionnez Créer des routines supplémentaires pour gérer les jeux de données ouverts.
- Vous pouvez définir le modèle à utiliser pour les noms de schéma dans le code converti. Pour Noms de schémas, choisissez l'une des options suivantes :
  - DB : utilise le nom de base de données SQL Server en tant que nom de schéma dans PostgreSQL.
  - SCHEMA : utilise le nom de schéma SQL Server en tant que nom de schéma dans PostgreSQL.
  - DB_SCHEMA : utilise une combinaison des noms de schéma et de base de données SQL Server en tant que nom de schéma dans PostgreSQL.
- Vous pouvez conserver la casse de vos noms d'objets sources. Pour éviter de convertir les noms d'objets en minuscules, sélectionnez Conserver les noms d'objets dans la même casse. Cette option s'applique uniquement lorsque vous activez l'option de sensibilité à la casse dans la base de données cible.
- Vous pouvez conserver les noms de paramètre de la base de données source. La conversion de schéma DMS peut ajouter des guillemets doubles aux noms des paramètres dans le code converti. Pour ce faire, sélectionnez Conserver les noms des paramètres d'origine.
- Vous pouvez conserver la longueur des paramètres de routine de votre base de données source. La conversion de schéma DMS crée des domaines et les utilise pour spécifier la longueur des paramètres de routine. Pour ce faire, sélectionnez Conserver la longueur des paramètres.

Paramètres de conversion de PostgreSQL vers MySQL

Les paramètres de conversion de PostgreSQL vers MySQL dans DMS Schema Conversion sont les suivants :

- Commentaires dans le code SQL converti : ce paramètre inclut des commentaires dans le code converti pour les actions dont le niveau de gravité est sélectionné ou supérieur. Ce paramètre prend en charge les valeurs suivantes :
  - Erreurs uniquement
  - Erreurs et avertissements
  - Tous les messages

Actualisation de vos schémas de base de données dans la conversion de schéma DMS

Après avoir créé un projet de migration, la conversion de schéma DMS stocke les informations relatives à vos schémas sources et cibles dans ce projet. La conversion de schéma DMS utilise le chargement différé pour charger les métadonnées uniquement lorsque cela est nécessaire, par exemple lorsque vous choisissez un nœud dans l'arborescence de la base de données. Vous pouvez utiliser le chargement rapide pour charger les informations du schéma plus rapidement. Pour ce faire, choisissez votre schéma, puis choisissez Charger des métadonnées dans Actions.

Une fois que vous avez chargé automatiquement ou manuellement l'objet dans votre projet de migration, la conversion de schéma DMS n'utilise plus le chargement différé. Ainsi, lorsque vous modifiez des objets, tels que des tables et des procédures dans la base de données, veillez à les actualiser dans votre projet de migration.

Pour actualiser les schémas de la base de données, sélectionnez les objets que vous souhaitez actualiser, puis choisissez Actualiser à partir de la base de données dans Actions. Vous pouvez actualiser les objets de base de données dans vos schémas de base de données sources et cibles :

- Source : si vous mettez à jour votre schéma de base de données source, choisissez Actualiser à partir de la base de données pour remplacer le schéma de votre projet par le dernier schéma de la base de données source.
- Cible : si vous mettez à jour le schéma de la base de données cible, la conversion de schéma DMS remplace le schéma de votre projet par le dernier schéma de la base de données cible. La conversion de schéma DMS remplace votre code converti par le code de la base de données

cible. Assurez-vous d'avoir appliqué le code converti à la base de données cible avant de choisir Actualiser à partir de la base de données. Sinon, convertissez à nouveau votre schéma de base de données source.

Enregistrement et application de votre code converti dans la conversion de schéma DMS

Une fois que la conversion de schéma DMS a converti vos objets de base de données source, il n'applique pas immédiatement le code converti à la base de données cible. Au lieu de cela, la conversion de schéma DMS stocke le code converti dans votre projet jusqu'à ce que vous soyez prêt à l'appliquer à la base de données cible.

Avant d'appliquer le code converti, vous pouvez mettre à jour le code de la base de données source et reconvertir les objets mis à jour pour traiter les éléments d'action existants. Pour plus d'informations sur les éléments que la conversion de schéma DMS ne peut pas convertir automatiquement, consultez Création de rapports d'évaluation de migration de base de données avec la conversion de schéma DMS. Pour plus d'informations sur l'actualisation de vos objets de base de données source dans le cadre d'un projet de migration pour la conversion de schéma DMS, consultez Actualisation de vos schémas de base de données.

Au lieu d'appliquer le code converti directement à la base de données dans la conversion de schéma DMS, vous pouvez enregistrer le code dans un fichier sous forme de script SQL. Vous pouvez consulter ces scripts SQL, les modifier si nécessaire, puis les appliquer manuellement à la base de données cible.

Enregistrement de votre code converti dans un fichier SQL

Vous pouvez enregistrer votre schéma converti sous forme de scripts SQL dans un fichier texte. Vous pouvez modifier le code converti pour traiter les éléments d'action que la conversion de schéma DMS ne peut pas convertir automatiquement. Vous pouvez ensuite exécuter vos scripts SQL mis à jour sur la base de données cible pour appliquer le code converti à la base de données cible.

Pour enregistrer votre schéma converti sous forme de scripts SQL

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration, puis choisissez Conversion de schéma.

4. Choisissez Lancer la conversion de schéma. La page Conversion de schéma s'ouvre.
5. Dans le volet de droite, choisissez le schéma de base de données cible ou sélectionnez les objets convertis que vous souhaitez enregistrer. Assurez-vous que la conversion de schéma DMS met en évidence le nom du nœud parent en bleu et active le menu Actions pour la base de données cible.
6. Choisissez Enregistrer en tant que SQL pour Actions. La boîte de dialogue Enregistrer s'affiche.
7. Choisissez Enregistrer en tant que SQL pour confirmer votre choix.

   La conversion de schéma DMS crée une archive avec les fichiers SQL et stocke cette archive dans votre compartiment Amazon S3.
8. (Facultatif) Changez le compartiment S3 de l'archive en modifiant les paramètres de conversion de schéma dans votre profil d'instance.
9. Ouvrez les scripts SQL à partir de votre compartiment S3.

Application de votre code converti

Une fois que vous êtes prêt à appliquer votre code converti à la base de données cible, choisissez les objets de base de données dans le volet de droite de votre projet. Vous pouvez appliquer des modifications à un schéma de base de données entier ou à certains objets de schéma de base de données.

Après avoir sélectionné les objets de base de données, la conversion de schéma DMS met en évidence le nom du nœud sélectionné ou du nœud parent en bleu. Il active ensuite le menu Actions. Choisissez Appliquer les modifications pour Actions. Dans la boîte de dialogue qui s'affiche, choisissez Appliquer pour confirmer votre choix et appliquer le code converti à la base de données cible.

Application du schéma du pack d'extension

Lorsque vous appliquez votre schéma converti à la base de données cible pour la première fois, la conversion de schéma DMS peut également appliquer le schéma du pack d'extension. Le schéma du pack d'extension émule les fonctions système de la base de données source nécessaires pour exécuter votre code converti pour la base de données cible. Si votre code converti utilise les fonctions du pack d'extension, assurez-vous d'appliquer le schéma du pack d'extension.

Pour appliquer manuellement le pack d'extension à la base de données cible, choisissez Appliquer les modifications pour Actions. Dans la boîte de dialogue qui s'affiche, choisissez Confirmer pour appliquer le pack d'extension à la base de données cible.

Nous vous recommandons de ne pas modifier le schéma du pack d'extension pour éviter des résultats inattendus dans le code converti.

Pour de plus amples informations, veuillez consulter Utilisation de packs d'extension dans la conversion de schéma DMS.

Utilisation de packs d'extension dans la conversion de schéma DMS

Dans la conversion de schéma DMS, un pack d'extension est un module complémentaire qui émule les fonctions de base de données source qui ne sont pas prises en charge dans la base de données cible. Utilisez un pack d'extension pour vous assurer que le code converti produit les mêmes résultats que le code source. Avant d'installer un pack d'extension, convertissez vos schémas de base de données.

Chaque pack d'extension inclut un schéma de base de données. Ce schéma inclut des fonctions, des procédures, des tables et des vues SQL permettant d'émuler certains objets de traitement des transactions en ligne (OLTP) ou des fonctions intégrées non prises en charge à partir de la base de données source.

Lorsque vous convertissez la base de données source, la conversion de schéma DMS ajoute un schéma supplémentaire à la base de données cible. Ce schéma met en œuvre les fonctions système SQL de la base de données source nécessaires pour exécuter votre code converti sur la base de données cible. Ce schéma supplémentaire est appelé schéma du kit d'extension.

Le schéma de kit d'extension est nommé selon votre base de données source, comme suit :

- Microsoft SQL Server – `aws_sqlserver_ext`
- Oracle – `aws_oracle_ext`

Vous pouvez appliquer les packs d'extension de deux façons :

- La conversion de schéma DMS peut appliquer automatiquement un pack d'extension lorsque vous appliquez votre code converti. La conversion de schéma DMS applique le pack d'extension avant d'appliquer tous les autres objets de schéma.

- Vous pouvez appliquer un pack d'extension manuellement. Pour ce faire, choisissez le schéma de pack d'extension dans l'arborescence de la base de données cible, puis choisissez Appliquer et Appliquer le pack d'extension.

Migration des bases de données vers leurs équivalents Amazon RDS avec AWS DMS

Les migrations de données homogènes dans AWS Database Migration Service (AWS DMS) simplifient la migration des bases de données autogérées sur site vers leurs équivalents Amazon Relational Database Service (Amazon RDS). Par exemple, vous pouvez utiliser des migrations de données homogènes pour migrer une base de données PostgreSQL sur site vers Amazon RDS for PostgreSQL ou Aurora PostgreSQL. Pour des migrations de données homogènes, AWS DMS utilise des outils de base de données natifs pour fournir des migrations comparables, simples et performantes.

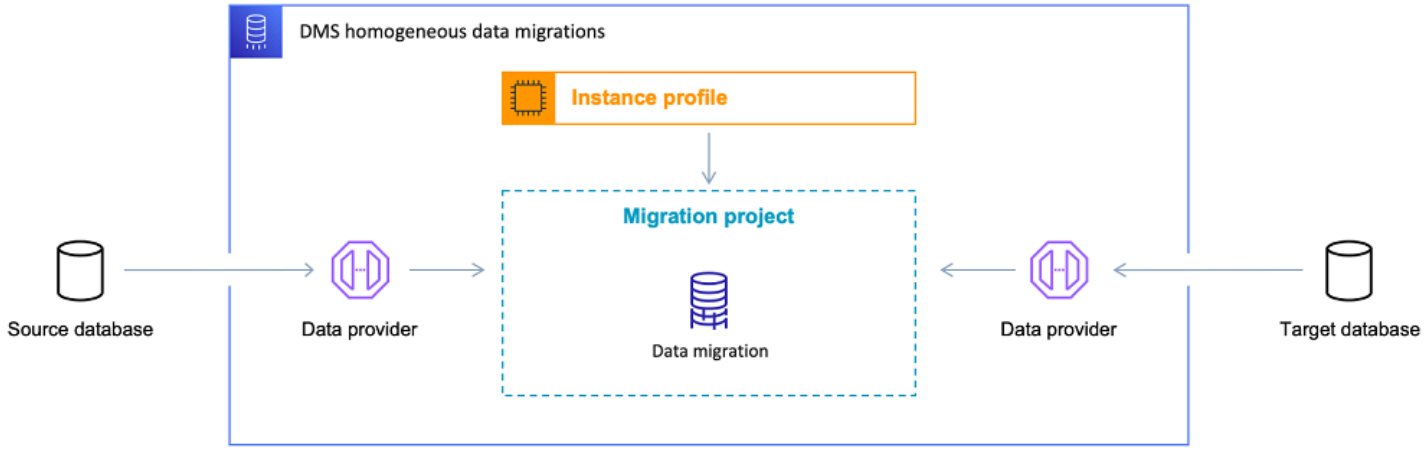
Les migrations de données homogènes sont sans serveur, ce qui signifie qu'AWS DMS met automatiquement à l'échelle les ressources nécessaires à votre migration. Avec des migrations de données homogènes, vous pouvez migrer des données, des partitions de table, des types de données et des objets secondaires tels que des fonctions, des procédures stockées, etc.

À un niveau élevé, les migrations de données homogènes fonctionnent avec des profils d'instance, des fournisseurs de données et des projets de migration. Lorsque vous créez un projet de migration avec les fournisseurs de données source et cible compatibles du même type, AWS DMS déploie un environnement sans serveur dans lequel s'exécute votre migration de données. Ensuite, AWS DMS se connecte au fournisseur de données source, lit les données sources, vide les fichiers sur le disque et restaure les données à l'aide des outils de base de données natifs. Pour plus d'informations sur les profils d'instance, les fournisseurs de données et les projets de migration, consultez Travailler avec des fournisseurs de données, des profils d'instance et des projets de migration dans AWS DMS.

Pour obtenir la liste des bases de données sources prises en charge, consultez Sources pour les migrations de données homogènes DMS.

Pour obtenir la liste des bases de données cibles prises en charge, consultez Cibles pour les migrations de données homogènes DMS.

Le diagramme suivant illustre le fonctionnement des migrations de données homogènes.



Les sections suivantes fournissent des informations sur l'utilisation des migrations de données homogènes.

Rubriques

- Régions AWS prises en charge
- Fonctionnalités
- Limitations liées aux migrations de données homogènes
- Vue d'ensemble du processus de migration des données homogène dans AWS DMS
- Configuration de migrations de données homogènes dans AWS DMS
- Création de fournisseurs de données sources pour des migrations de données homogènes dans AWS DMS
- Création de fournisseurs de données cibles pour les migrations de données homogènes dans AWS DMS
- Exécution de migrations de données homogènes dans AWS DMS
- Résolution des problèmes liés aux migrations de données homogènes dans AWS DMS

Régions AWS prises en charge

Vous pouvez exécuter des migrations de données homogènes dans les Régions AWS suivantes.

Nom de la région	Région
USA Est (Virginie du Nord)	us-east-1

Nom de la région	Région
USA Est (Ohio)	us-east-2
USA Ouest (Oregon)	us-west-2
Asia Pacific (Tokyo)	ap-northeast-1
Asie-Pacifique (Singapour)	ap-southeast-1
Asie-Pacifique (Sydney)	ap-southeast-2
Europe (Francfort)	eu-central-1
Europe (Stockholm)	eu-north-1
Europe (Irlande)	eu-west-1

Fonctionnalités

Les migrations de données homogènes fournissent les fonctionnalités suivantes :

- AWS DMS gère automatiquement les ressources de calcul et de stockage du AWS Cloud qui sont requises pour les migrations de données homogènes. AWS DMS déploie ces ressources dans un environnement sans serveur lorsque vous lancez une migration de données.
- AWS DMS utilise des outils de base de données natifs pour lancer une migration entièrement automatisée entre les bases de données du même type.
- Vous pouvez utiliser des migrations de données homogènes pour migrer vos données et les objets secondaires tels que les partitions, les fonctions, les procédures stockées, etc.
- Vous pouvez exécuter des migrations de données homogènes dans les trois modes de migration suivants : chargement complet, réplication continue et chargement complet avec réplication continue.
- Pour les migrations de données homogènes, vous pouvez utiliser des bases de données sur site, Amazon EC2 ou Amazon RDS en tant que source. Vous pouvez choisir Amazon RDS ou Amazon Aurora en tant que cible de migration pour les migrations de données homogènes.

Limitations liées aux migrations de données homogènes

Les limitations suivantes s'appliquent lors de l'utilisation des migrations de données homogènes :

- Les migrations de données homogènes ne prennent pas en charge les règles de sélection. De plus, vous ne pouvez pas utiliser des règles de transformation pour changer le type de données des colonnes, déplacer des objets d'un schéma vers un autre ou modifier les noms des objets.
- Les migrations de données homogènes ne fournissent pas d'outil intégré pour la validation des données.
- Lorsque vous utilisez des migrations de données homogènes avec PostgreSQL, AWS DMS migre les vues sous forme de tables vers la base de données cible.
- Les migrations de données homogènes ne capturent pas les modifications au niveau du schéma lors d'une réplication de données continue. Si vous créez une nouvelle table dans la base de données source, AWS DMS ne peut pas migrer cette table. Pour migrer cette nouvelle table, relancez votre migration de données.
- Vous ne pouvez pas utiliser de migrations de données homogènes dans AWS DMS pour migrer des données d'une version de base de données supérieure vers une version de base de données inférieure.
- Vous ne pouvez pas utiliser de migrations de données homogènes dans l'interface CLI ou l'API.
- Les migrations de données homogènes ne permettent pas d'établir une connexion avec des instances de base de données dans les plages d'adresses CIDR secondaires d'un VPC.
- Vous ne pouvez pas utiliser le port 8081 pour effectuer des migrations homogènes à partir de vos fournisseurs de données.

Vue d'ensemble du processus de migration des données homogène dans AWS DMS

Vous pouvez utiliser les migrations de données homogènes dans AWS DMS pour migrer les données entre deux bases de données du même type. Utilisez le flux de travail suivant pour créer et exécuter une migration de données.

1. Créez la politique et le rôle AWS Identity and Access Management (IAM) requis. Pour de plus amples informations, veuillez consulter Création de ressources IAM.
2. Configurez vos bases de données sources et cibles et créez des utilisateurs de base de données dotés des autorisations minimales requises pour les migrations de données homogènes dans

AWS DMS. Pour plus d'informations, consultez Création de fournisseurs de données sources et Création de fournisseurs de données cibles.

3. Stockez vos informations d'identification de base de données source et cible dans AWS Secrets Manager. Pour plus d'informations, consultez Étape 1 : Créer le secret dans le Guide de l'utilisateur AWS Secrets Manager.
4. Créez un groupe de sous-réseaux, un profil d'instance et des fournisseurs de données dans la console AWS DMS. Pour plus d'informations, consultez Création d'un groupe de sous-réseaux, Création de profils d'instance et Création de fournisseurs de données.
5. Créez un projet de migration à l'aide des ressources que vous avez créées à l'étape précédente. Pour de plus amples informations, veuillez consulter Création de projets de migration.
6. Créez, configurez et démarrez une migration de données. Pour de plus amples informations, veuillez consulter Création d'une migration de données.
7. Après avoir terminé le chargement complet ou la réplication continue, vous pouvez commencer à utiliser votre nouvelle base de données cible.
8. Nettoyez vos ressources. Amazon met fin à la migration des données de votre projet de migration dans les trois jours suivant la fin de la migration. Toutefois, vous devez supprimer manuellement les ressources telles que le profil d'instance, les fournisseurs de données, la politique et le rôle IAM, ainsi que les secrets dans AWS Secrets Manager.

Pour plus d'informations sur les migrations de données homogènes dans AWS DMS, consultez la procédure de migration étape par étape pour les migrations de PostgreSQL vers Amazon RDS for PostgreSQL.

Cette vidéo présente les migrations de données homogènes dans AWS DMS et vous aide à vous familiariser avec cette fonctionnalité.

Configuration de migrations de données homogènes dans AWS DMS

Pour configurer des migrations de données homogènes dans AWS DMS, effectuez les tâches prérequises suivantes.

Rubriques

- Création des ressources IAM requises pour les migrations de données homogènes dans AWS DMS

- Configuration d'un réseau pour les migrations de données homogènes dans AWS DMS

Création des ressources IAM requises pour les migrations de données homogènes dans AWS DMS

Pour exécuter des migrations de données homogènes, vous devez créer une politique IAM et un rôle IAM dans votre compte afin d'interagir avec d'autres services AWS. Dans cette section, vous allez créer ces ressources IAM requises.

Rubriques

- Création d'une politique IAM pour les migrations de données homogènes dans AWS DMS
- Création d'un rôle IAM pour les migrations de données homogènes dans AWS DMS

Création d'une politique IAM pour les migrations de données homogènes dans AWS DMS

Pour accéder à vos bases de données et migrer les données, AWS DMS crée un environnement sans serveur pour les migrations de données homogènes. Dans cet environnement, AWS DMS nécessite un accès à l'appairage de VPC, aux tables de routage, aux groupes de sécurité et à d'autres ressources AWS. AWS DMS stocke également les journaux, les métriques et la progression de chaque migration de données dans Amazon CloudWatch. Pour créer un projet de migration de données, AWS DMS a besoin d'accéder à ces services.

Au cours de cette étape, vous créez une politique IAM qui fournit à AWS DMS un accès aux ressources Amazon EC2 et CloudWatch. Ensuite, créez un rôle IAM et attachez cette politique.

Pour créer une politique IAM pour les migrations de données homogènes dans AWS DMS

1. Connectez-vous à la AWS Management Console et ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Policies (Politiques).
3. Sélectionnez Créer une politique.
4. Sur la page Créer une stratégie, choisissez l'onglet JSON.
5. Collez le code JSON suivant dans l'éditeur.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeVpcs",
                "ec2:DescribePrefixLists",
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "servicequotas:GetServiceQuota"
            ],
            "Resource": "arn:aws:servicequotas:*:*:vpc/L-0EA8095F"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:dms-data-migration-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:dms-data-migration-*:log-
stream:dms-data-migration-*"
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*"
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "ec2:CreateRoute",
                "ec2:DeleteRoute"
            ],
            "Resource": "arn:aws:ec2:*:*:route-table/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:security-group-rule/*",
                "arn:aws:ec2:*:*:route-table/*",
                "arn:aws:ec2:*:*:vpc-peering-connection/*",
                "arn:aws:ec2:*:*:vpc/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group-rule/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AcceptVpcPeeringConnection",
                "ec2:ModifyVpcPeeringConnectionOptions"
            ],
```

```
            "Resource": "arn:aws:ec2:*:*:vpc-peering-connection/*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:AcceptVpcPeeringConnection",
            "Resource": "arn:aws:ec2:*:*:vpc/*"
        }
    ]
}
```

6. Choisissez Next: Tags (Suivant : Balises), puis Next: Review (Suivant : Vérification).
7. Entrez **`HomogeneousDataMigrationsPolicy`** pour Nom*, puis choisissez Créer une politique.

Création d'un rôle IAM pour les migrations de données homogènes dans AWS DMS

Au cours de cette étape, vous créez un rôle IAM qui fournit à AWS DMS un accès à AWS Secrets Manager, Amazon EC2 et CloudWatch.

Pour créer un rôle IAM pour les migrations de données homogènes dans AWS DMS

1. Connectez-vous à la AWS Management Console, puis ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Roles (Rôles).
3. Sélectionnez Create role (Créer un rôle).
4. Sur la page Sélectionner une entité de confiance, pour Type d'entité approuvée, choisissez Service AWS. Pour Cas d'utilisation d'autres services AWS, choisissez DMS.
5. Cochez la case DMS et choisissez Suivant.
6. Sur la page Ajouter des autorisations, choisissez l'autorisation HomogeneousDataMigrationsPolicy que vous venez de créer. Choisissez également SecretsManagerReadWrite. Choisissez Next (Suivant).
7. Sur la page Nommer, vérifier et créer, entrez **`HomogeneousDataMigrationsRole`** pour Nom du rôle et choisissez Créer un rôle.
8. Sur la page Rôles, entrez **`HomogeneousDataMigrationsRole`** pour Nom du rôle. Choisissez HomogeneousDataMigrationsRole.
9. Sur la page HomogeneousDataMigrationsRole, cliquez sur l'onglet Relations d'approbation. Choisissez Edit trust policy (Modifier la politique).

10. Sur la page Modifier la politique d'approbation, collez le code JSON suivant dans l'éditeur en remplaçant le texte existant.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "dms-data-migrations.amazonaws.com",
                    "dms.your_region.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

    Dans l'exemple précédent, remplacez *your_region* par le nom de votre Région AWS.

    La politique précédente basée sur les ressources fournit aux principaux de service AWS DMS les autorisations nécessaires pour effectuer des tâches conformément à la politique SecretsManagerReadWrite gérée par AWS et à la politique HomogeneousDataMigrationsPolicy gérée par le client.

11. Choisissez Update policy (Mettre à jour une politique).

Configuration d'un réseau pour les migrations de données homogènes dans AWS DMS

AWS DMS crée un environnement sans serveur pour les migrations de données homogènes dans un cloud privé virtuel (VPC) basé sur le service Amazon VPC. Lorsque vous créez votre profil d'instance, vous spécifiez le VPC à utiliser. Vous pouvez utiliser votre VPC par défaut pour votre compte et la Région AWS, ou vous pouvez créer un nouveau VPC.

Pour chaque migration de données, AWS DMS établit une connexion d'appairage de VPC avec le VPC que vous utilisez pour votre profil d'instance. Ensuite, AWS DMS ajoute le bloc d'adresses CIDR dans le groupe de sécurité associé à votre profil d'instance. Étant donné qu'AWS DMS attache une

adresse IP publique à votre profil d'instance, toutes vos migrations de données qui utilisent le même profil d'instance ont la même adresse IP publique. Lorsque votre migration de données s'arrête ou échoue, AWS DMS supprime la connexion d'appairage de VPC.

Pour éviter que le bloc d'adresses CIDR ne chevauche le VPC de votre profil d'instance, AWS DMS utilise le préfixe `/24` de l'un des blocs d'adresses CIDR suivants : `10.0.0.0/8`, `172.16.0.0/12` et `192.168.0.0/16`. Par exemple, si vous exécutez trois migrations de données en parallèle, AWS DMS utilise les blocs d'adresses CIDR suivants pour établir une connexion d'appairage de VPC.

- `192.168.0.0/24` : pour la première migration de données
- `192.168.1.0/24` : pour la deuxième migration de données
- `192.168.2.0/24` : pour la troisième migration de données

Vous pouvez utiliser différentes configurations réseau pour configurer l'interaction entre vos bases de données sources et cibles avec AWS DMS. En outre, pour une réplication continue des données, vous devez configurer l'interaction entre vos bases de données sources et cibles. Ces configurations dépendent de l'emplacement de votre fournisseur de données source et de vos paramètres réseau. Les sections suivantes décrivent les configurations réseau courantes.

Rubriques

- Utilisation d'un VPC unique pour les fournisseurs de données source et cible
- Utilisation de différents VPC pour les fournisseurs de données sources et cibles
- Utilisation d'un fournisseur de données source sur site
- Configuration de la réplication continue des données

Utilisation d'un VPC unique pour les fournisseurs de données source et cible

Dans cette configuration, AWS DMS se connecte à vos fournisseurs de données sources et cibles au sein du réseau privé.

Pour configurer un réseau lorsque vos fournisseurs de données sources et cibles se trouvent dans le même VPC

1. Créez le groupe de sous-réseaux dans la console AWS DMS avec le VPC et les sous-réseaux utilisés par vos fournisseurs de données sources et cibles. Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux.

2. Créez le profil d'instance dans la console AWS DMS avec le VPC et le groupe de sous-réseaux que vous avez créés. Choisissez également les groupes de sécurité de VPC utilisés par vos fournisseurs de données sources et cibles. Pour de plus amples informations, veuillez consulter Création de profils d'instance.

Cette configuration ne vous oblige pas à utiliser l'adresse IP publique pour les migrations de données.

Utilisation de différents VPC pour les fournisseurs de données sources et cibles

Dans cette configuration, AWS DMS utilise un réseau privé pour se connecter à votre fournisseur de données source ou cible. Pour un autre fournisseur de données, AWS DMS utilise un réseau public. En fonction du fournisseur de données qui se trouve dans le même VPC que votre profil d'instance, choisissez l'une des configurations suivantes.

Pour configurer un réseau privé pour votre fournisseur de données source et un réseau public pour votre fournisseur de données cible

1. Créez le groupe de sous-réseaux dans la console AWS DMS avec le VPC et les sous-réseaux utilisés par votre fournisseur de données source. Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux.
2. Créez le profil d'instance dans la console AWS DMS avec le VPC et le groupe de sous-réseaux que vous avez créés. Choisissez également les groupes de sécurité de VPC utilisés par votre fournisseur de données source. Pour de plus amples informations, veuillez consulter Création de profils d'instance.
3. Ouvrez votre projet de migration. Dans l'onglet Migrations de données, choisissez votre migration de données. Notez l'adresse IP publique sous Connectivité et sécurité dans l'onglet Détails.
4. Autorisez l'accès à partir de l'adresse IP publique de votre migration de données dans le groupe de sécurité de la base de données cible. Pour plus d'informations, consultez Contrôle d'accès par groupe de sécurité dans le Guide de l'utilisateur Amazon Relational Database Service.

Pour configurer un réseau public pour votre fournisseur de données source et un réseau privé pour votre fournisseur de données cible

1. Créez le groupe de sous-réseaux dans la console AWS DMS avec le VPC et les sous-réseaux utilisés par votre fournisseur de données cible. Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux.
2. Créez le profil d'instance dans la console AWS DMS avec le VPC et le groupe de sous-réseaux que vous avez créés. Choisissez également les groupes de sécurité de VPC utilisés par votre fournisseur de données cible. Pour de plus amples informations, veuillez consulter Création de profils d'instance.
3. Ouvrez votre projet de migration. Dans l'onglet Migrations de données, choisissez votre migration de données. Notez l'adresse IP publique sous Connectivité et sécurité dans l'onglet Détails.
4. Autorisez l'accès à partir de l'adresse IP publique de votre migration de données dans le groupe de sécurité de la base de données source. Pour plus d'informations, consultez Contrôle d'accès par groupe de sécurité dans le Guide de l'utilisateur Amazon Relational Database Service.

Utilisation d'un fournisseur de données source sur site

Dans cette configuration, AWS DMS se connecte à votre fournisseur de données source au sein du réseau public. AWS DMS utilise un réseau privé pour se connecter à votre fournisseur de données cible.

Pour configurer un réseau pour votre fournisseur de données source sur site

1. Créez le groupe de sous-réseaux dans la console AWS DMS avec le VPC et les sous-réseaux utilisés par votre fournisseur de données cible. Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux.
2. Créez le profil d'instance dans la console AWS DMS avec le VPC et le groupe de sous-réseaux que vous avez créés. Choisissez également les groupes de sécurité de VPC utilisés par votre fournisseur de données cible. Pour de plus amples informations, veuillez consulter Création de profils d'instance.
3. Ouvrez votre projet de migration. Dans l'onglet Migrations de données, choisissez votre migration de données. Notez l'adresse IP publique sous Connectivité et sécurité dans l'onglet Détails.

4. Autorisez l'accès à la base de données source à partir de l'adresse IP publique de votre migration de données dans AWS DMS.

AWS DMS crée des règles entrantes ou sortantes dans les groupes de sécurité de VPC. Assurez-vous de ne pas supprimer ces règles, car cette action peut entraîner l'échec de la migration de vos données. Vous pouvez configurer vos propres règles dans les groupes de sécurité de VPC. Nous vous recommandons d'ajouter une description à vos règles afin de pouvoir les gérer.

Configuration de la réplication continue des données

Pour exécuter des migrations de données de type Capture des données de chargement complet et de modification (CDC) ou Capture de données modifiées (CDC), vous devez autoriser la connexion entre vos bases de données sources et cibles.

Pour configurer une connexion entre vos bases de données sources et cibles accessibles en public

1. Prenez note des adresses IP publiques de vos bases de données sources et cibles.
2. Autorisez l'accès à la base de données source à partir de l'adresse IP publique de la base de données cible.
3. Autorisez l'accès à la base de données cible à partir de l'adresse IP publique de la base de données source.

Pour configurer une connexion entre vos bases de données sources et cibles accessibles en privé dans un VPC unique

1. Prenez note des adresses IP privées de vos bases de données sources et cibles.

   > **Important**
   >
   > Si vos bases de données sources et cibles se trouvent dans des VPC ou des réseaux différents, vous ne pouvez utiliser que des adresses IP publiques pour vos bases de données sources et cibles. Vous ne pouvez utiliser que des noms d'hôtes ou des adresses IP publics dans les fournisseurs de données.

2. Autorisez l'accès à la base de données source à partir de l'adresse IP privée de la base de données cible.

3. Autorisez l'accès à la base de données cible à partir de l'adresse IP privée de la base de données source.

Création de fournisseurs de données sources pour des migrations de données homogènes dans AWS DMS

Vous pouvez utiliser des bases de données compatibles MySQL, PostgreSQL et MongoDB comme fournisseur de données source pour in. Migrations de données homogènes AWS DMS

Pour les versions de base de données prises en charge, consultezFournisseurs de données sources pour les migrations de données homogènes DMS.

Votre fournisseur de données source peut être une base de données sur site, Amazon EC2 ou Amazon RDS.

Rubriques

- Utilisation d'une base de données compatible MySQL comme source pour des migrations de données homogènes dans AWS DMS
- Utilisation d'une base de données PostgreSQL comme source pour des migrations de données homogènes dans AWS DMS
- Utilisation d'une base de données compatible MongoDB comme source pour des migrations de données homogènes dans AWS DMS

Utilisation d'une base de données compatible MySQL comme source pour des migrations de données homogènes dans AWS DMS

Vous pouvez utiliser une base de données compatible MySQL (MySQL ou MariaDB) en tant que source pour Migrations de données homogènes dans AWS DMS. Dans ce cas, votre fournisseur de données source peut être une base de données sur site, Amazon EC2, RDS for MySQL ou MariaDB.

Pour exécuter des migrations de données homogènes, vous devez utiliser un utilisateur de base de données disposant des privilèges `SELECT` pour toutes les tables sources et tous les objets secondaires à répliquer. Pour les tâches de capture des données de modification (CDC), cet utilisateur doit également disposer des privilèges `REPLICATION CLIENT` (`BINLOG MONITOR` pour les versions de MariaDB ultérieures à 10.5.2) et `REPLICATION SLAVE`. Pour une migration de données de chargement complet, vous n'avez pas besoin de ces deux privilèges.

Utilisez le script suivant pour créer un utilisateur de base de données doté des autorisations requises dans la base de données MySQL. Exécutez les GRANT requêtes pour toutes les bases de données vers lesquelles vous migrez AWS.

```
CREATE USER 'your_user'@'%' IDENTIFIED BY 'your_password';

GRANT REPLICATION SLAVE, REPLICATION CLIENT  ON *.* TO 'your_user'@'%';
GRANT SELECT, RELOAD, LOCK TABLES, SHOW VIEW, EVENT, TRIGGER ON *.* TO 'your_user'@'%';

GRANT BACKUP_ADMIN ON *.* TO 'your_user'@'%';
```

Dans l'exemple précédent, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations. Si la version de la base de données MySQL source est antérieure à 8.0, vous pouvez ignorer la commande GRANT BACKUP_ADMIN.

Utilisez le script suivant pour créer un utilisateur de base de données doté des autorisations requises dans la base de données MariaDB. Exécutez les requêtes GRANT pour toutes les bases de données vers lesquelles vous migrez AWS.

```
CREATE USER 'your_user'@'%' IDENTIFIED BY 'your_password';
GRANT SELECT, RELOAD, LOCK TABLES, REPLICATION SLAVE, BINLOG MONITOR, SHOW VIEW ON  *.*
 TO 'your_user'@'%';
```

Dans l'exemple précédent, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

Les sections suivantes décrivent les prérequis de configuration spécifiques pour les bases de données MySQL autogérées et gérées par AWS.

Rubriques

- Utilisation d'une base de données compatible MySQL autogérée en tant que source pour les migrations de données homogènes
- Utilisation d'une base de données compatible MySQL AWS gérée comme source pour des migrations de données homogènes dans AWS DMS
- Limites d'utilisation d'une base de données compatible MySQL en tant que source pour une migration de données homogène

Utilisation d'une base de données compatible MySQL autogérée en tant que source pour les migrations de données homogènes

Cette section explique comment configurer vos bases de données compatibles MySQL hébergées sur site ou sur des instances Amazon EC2.

Choisissez la version de la base de données MySQL ou MariaDB source. Assurez-vous qu'elle AWS DMS prend en charge la version de votre base de données MySQL ou MariaDB source, comme décrit dans. Sources pour les migrations de données homogènes DMS

Pour utiliser la CDC, assurez-vous d'activer la journalisation binaire. Pour activer la journalisation binaire, configurez les paramètres suivants dans le fichier `my.ini` (Windows) ou `my.cnf` (UNIX) de la base de données MySQL ou MariaDB.

Paramètre	Valeur
`server-id`	Définissez ce paramètre à une valeur 1 ou supérieure.
`log-bin`	Définissez le chemin d'accès au fichier journal binaire, par exemple `log-bin=E:\MySql_Logs\BinLog` . N'ajoutez pas d'extension de fichier.
`binlog_format`	Définissez ce paramètre à `ROW`. Nous recommandons d'utiliser ce paramètre lors de la réplication car, dans certains cas, lorsque `binlog_format` est défini sur `STATEMENT` , cela peut entraîner des incohérences lors de la réplication des données sur la cible. Le moteur de base de données écrit également des données incohérentes similaires sur la cible lorsque `binlog_format` est défini sur `MIXED`, car le moteur de base de données bascule automatiquement vers la journalisation basée sur `STATEMENT` .
`expire_logs_days`	Définissez ce paramètre à une valeur 1 ou supérieure. Pour éviter l'utilisation excessive d'espace disque, nous recommandons de ne pas utiliser la valeur par défaut de 0.
`binlog_checksum`	Définissez ce paramètre à `NONE`.
`binlog_row_image`	Définissez ce paramètre à `FULL`.

Paramètre	Valeur
`log_slave_updates`	Définissez ce paramètre sur `TRUE` si vous utilisez un réplica MySQL ou MariaDB en tant que source.

Utilisation d'une base de données compatible MySQL AWS gérée comme source pour des migrations de données homogènes dans AWS DMS

Cette section explique comment configurer vos instances de base de données Amazon RDS for MySQL et Amazon RDS for MariaDB.

Lorsque vous utilisez une base de données MySQL ou MariaDB AWS gérée comme source pour des migrations de données homogènes, assurez-vous AWS DMS de remplir les conditions préalables suivantes pour CDC :

- Pour activer les journaux binaires pour RDS for MySQL et pour MariaDB, activez les sauvegardes automatiques au niveau de l'instance. Pour activer les journaux binaires pour un cluster Aurora MySQL, modifiez la variable `binlog_format` dans le groupe de paramètres. Vous n'avez pas besoin d'activer les sauvegardes automatiques pour un cluster Aurora MySQL.

  Définissez ensuite le paramètre `binlog_format` sur `ROW`.

  Pour plus d'informations sur la configuration des sauvegardes automatiques, consultez Activation des sauvegardes automatiques dans le Guide de l'utilisateur Amazon RDS.

  Pour plus d'informations sur la configuration de la journalisation binaire pour une base de données Amazon RDS for MySQL ou MariaDB, consultez Configuration de la journalisation binaire MySQL dans le Guide de l'utilisateur Amazon RDS.

  Pour plus d'informations sur la configuration de la journalisation binaire pour un cluster Aurora MySQL, consultez Comment puis-je activer la journalisation binaire pour mon cluster Amazon Aurora MySQL ?
- Assurez-vous que les journaux binaires sont disponibles pour AWS DMS. Étant donné que les bases de données MySQL et MariaDB AWS gérées par -managed purgent les journaux binaires dès que possible, vous devez augmenter la durée pendant laquelle les journaux restent disponibles. Par exemple, pour accroître la rétention des journaux à 24 heures, exécutez la commande suivante.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

- Définissez le paramètre `binlog_row_image` sur `Full`.
- Définissez le paramètre `binlog_checksum` sur `NONE`.
- Si vous utilisez un réplica Amazon RDS MySQL ou MariaDB en tant que source, activez les sauvegardes sur le réplica en lecture et assurez-vous de définir le paramètre `log_slave_updates` sur `TRUE`.

Limites d'utilisation d'une base de données compatible MySQL en tant que source pour une migration de données homogène

Les limites suivantes s'appliquent à l'utilisation d'une base de données compatible MySQL en tant que source pour une migration de données homogène :

- Les objets MariaDB tels que les séquences ne sont pas pris en charge dans les tâches de migration homogène.
- La migration de MariaDB vers Amazon RDS MySQL/Aurora MySQL peut échouer en raison de différences entre objets incompatibles.
- Le nom d'utilisateur que vous utilisez pour vous connecter à votre source de données présente les limites suivantes :
  - Il peut comporter de 2 à 64 caractères.
  - Il ne peut pas comporter d'espaces.
  - Il peut inclure les caractères suivants : a-z, A-Z, 0-9, trait de soulignement (_).
  - Il doit commencer par a-z ou A-Z.
- Le mot de passe que vous utilisez pour vous connecter à votre source de données présente les limites suivantes :
  - Il peut comporter de 1 à 128 caractères.
  - Il ne peut contenir aucun des éléments suivants : guillemet simple (’), guillemet double ("), point-virgule (;) ou espace.

Utilisation d'une base de données PostgreSQL comme source pour des migrations de données homogènes dans AWS DMS

Vous pouvez utiliser une base de données PostgreSQL en tant que source pour Migrations de données homogènes dans AWS DMS. Dans ce cas, votre fournisseur de données source peut être une base de données sur site, Amazon EC2 ou RDS for PostgreSQL.

Pour exécuter des migrations de données homogènes, accordez des autorisations de superutilisateur à l'utilisateur de base de données que vous avez spécifié AWS DMS pour votre base de données source PostgreSQL. L'utilisateur de base de données a besoin d'autorisations de superutilisateur pour accéder aux fonctions spécifiques à la réplication dans la source. Pour une migration de données de chargement complet, votre utilisateur de base de données a besoin d'autorisations `SELECT` sur les tables pour les migrer.

Utilisez le script suivant pour créer un utilisateur de base de données doté des autorisations requises dans la base de données source PostgreSQL. Exécutez la `GRANT` requête pour toutes les bases de données vers lesquelles vous migrez AWS.

```
CREATE USER your_user WITH LOGIN PASSWORD 'your_password';
ALTER USER your_user WITH SUPERUSER;
GRANT SELECT ON ALL TABLES IN SCHEMA schema_name TO your_user;
```

Dans l'exemple précédent, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

Les sections suivantes décrivent les prérequis de configuration spécifiques pour les bases de données PostgreSQL autogérées et gérées par AWS.

Rubriques

- Utilisation d'une base de données PostgreSQL autogérée comme source pour des migrations de données homogènes dans AWS DMS
- Utilisation d'une base de données PostgreSQL AWS gérée comme source pour des migrations de données homogènes dans AWS DMS
- Limites d'utilisation d'une base de données compatible PostgreSQL en tant que source pour une migration de données homogène

Utilisation d'une base de données PostgreSQL autogérée comme source pour des migrations de données homogènes dans AWS DMS

Cette section explique comment configurer vos bases de données PostgreSQL hébergées sur site ou sur des instances Amazon EC2.

Vérifiez la version de la base de données PostgreSQL source. Assurez-vous qu'elle est AWS DMS compatible avec la version de votre base de données PostgreSQL source, comme décrit dans. Sources pour les migrations de données homogènes DMS

Les migrations de données homogènes prennent en charge la capture des données de modification (CDC) à l'aide de la réplication logique. Pour activer la réplication logique sur une base de données source PostgreSQL autogérée, définissez les paramètres suivants dans le fichier de configuration `postgresql.conf` :

- Définissez `wal_level` sur `logical`.
- Définissez une valeur supérieure à 1 pour `max_replication_slots`.

  Définissez la valeur `max_replication_slots` selon le nombre de tâches que vous souhaitez exécuter. Par exemple, pour exécuter cinq tâches, vous définissez au moins cinq emplacements. Les emplacements s'ouvrent automatiquement dès qu'une tâche commence et restent ouvert même lorsque la tâche n'est plus en cours d'exécution. Assurez-vous de supprimer manuellement les emplacements ouverts.
- Définissez une valeur supérieure à 1 pour `max_wal_senders`.

  Le paramètre `max_wal_senders`définit le nombre de tâches simultanées qui peuvent s'exécuter.
- Le paramètre `wal_sender_timeout` met fin aux connexions de réplication qui sont inactives plus longtemps que le nombre de millisecondes spécifié. La valeur par défaut est de 60 000 millisecondes (60 secondes). La définition de la valeur sur 0 (zéro) désactive le mécanisme d'expiration et constitue une valeur valide pour DMS.

Certains paramètres sont statiques et vous ne pouvez les définir qu'au démarrage du serveur. Toute modification apportée à leurs entrées dans le fichier de configuration est ignorée tant que le serveur n'est pas redémarré. Pour de plus amples informations, veuillez consulter la documentation sur PostgreSQL.

Utilisation d'une base de données PostgreSQL AWS gérée comme source pour des migrations de données homogènes dans AWS DMS

Cette section explique comment configurer vos instances de base de données Amazon RDS for PostgreSQL.

Utilisez le compte utilisateur AWS principal de l'instance de base de données PostgreSQL comme compte utilisateur du fournisseur de données sources PostgreSQL pour des migrations de données homogènes dans. AWS DMS Le compte d'utilisateur principal a les rôles nécessaires qui lui permettent de configurer la capture des données modifiées. Si vous utilisez un compte autre que le compte d'utilisateur principal, ce compte doit avoir le rôle `rds_superuser` et le rôle `rds_replication`. Le rôle `rds_replication` accorde les autorisations permettant de gérer des emplacements logiques et de diffuser les données à l'aide d'emplacements logiques.

Utilisez l'exemple de code suivant pour octroyer les rôles `rds_superuser` et `rds_replication`.

```
GRANT rds_superuser to your_user;
GRANT rds_replication to your_user;
```

Dans l'exemple précédent, remplacez *`your_user`* par le nom de votre utilisateur de base de données.

Pour activer la réplication logique, définissez le paramètre `rds.logical_replication` de votre groupe de paramètres de base de données sur 1. Ce paramètre statique nécessite un redémarrage de l'instance DB pour son application.

Limites d'utilisation d'une base de données compatible PostgreSQL en tant que source pour une migration de données homogène

Les limites suivantes s'appliquent à l'utilisation d'une base de données compatible PostgreSQL en tant que source pour une migration de données homogène :

- Le nom d'utilisateur que vous utilisez pour vous connecter à votre source de données présente les limites suivantes :
  - Il peut comporter de 2 à 64 caractères.
  - Il ne peut pas comporter d'espaces.
  - Il peut inclure les caractères suivants : a-z, A-Z, 0-9, trait de soulignement (_).
  - Il doit commencer par a-z ou A-Z.

- Le mot de passe que vous utilisez pour vous connecter à votre source de données présente les limites suivantes :
  - Il peut comporter de 1 à 128 caractères.
  - Il ne peut contenir aucun des éléments suivants : guillemet simple (’), guillemet double ("), point-virgule (;) ou espace.

Utilisation d'une base de données compatible MongoDB comme source pour des migrations de données homogènes dans AWS DMS

Vous pouvez utiliser une base de données compatible MongoDB comme source pour les migrations de données homogènes dans. AWS DMS Dans ce cas, votre fournisseur de données source peut être une base de données Amazon EC2 pour MongoDB sur site ou une base de données Amazon DocumentDB (compatible avec MongoDB).

Pour les versions de base de données prises en charge, consultezFournisseurs de données sources pour les migrations de données homogènes DMS.

Les sections suivantes décrivent les prérequis de configuration spécifiques pour les bases de données MongoDB autogérées et les bases de données AWS Amazon DocumentDB gérées par des tiers.

Rubriques

- Utilisation d'une base de données MongoDB autogérée comme source pour des migrations de données homogènes dans AWS DMS
- Utilisation d'une base de données Amazon DocumentDB comme source pour des migrations de données homogènes dans AWS DMS
- Fonctionnalités d'utilisation d'une base de données compatible MongoDB comme source pour des migrations de données homogènes
- Limitations liées à l'utilisation d'une base de données compatible avec MongoDB comme source pour des migrations de données homogènes
- Bonnes pratiques d'utilisation d'une base de données compatible avec MongoDB comme source pour des migrations de données homogènes

Utilisation d'une base de données MongoDB autogérée comme source pour des migrations de données homogènes dans AWS DMS

Cette section décrit comment configurer vos bases de données MongoDB hébergées sur site ou sur des instances Amazon EC2.

Vérifiez la version de votre base de données MongoDB source. Assurez-vous qu'elle AWS DMS prend en charge la version de votre base de données MongoDB source, comme décrit dans. Fournisseurs de données sources pour les migrations de données homogènes DMS

Pour effectuer des migrations de données homogènes avec une source MongoDB, vous pouvez créer soit un compte utilisateur avec des privilèges root, soit un utilisateur autorisé uniquement sur la base de données à migrer. Pour plus d'informations sur la création d'utilisateurs, consultezAutorisations nécessaires lors de l'utilisation de MongoDB comme source pour AWS DMS.

Pour utiliser la réplication continue ou le CDC avec MongoDB, AWS DMS il faut accéder au journal des opérations MongoDB (oplog). Pour de plus amples informations, veuillez consulter Configuration d'un ensemble de réplicas MongoDB pour la CDC.

Pour plus d'informations sur les méthodes d'authentification MongoDB, consultez. Exigences de sécurité lors de l'utilisation de MongoDB comme source pour AWS DMS

Pour MongoDB en tant que source, les migrations de données homogènes prennent en charge tous les types de données pris en charge par Amazon DocumentDB.

Pour MongoDB en tant que source, pour stocker les informations d'identification des utilisateurs dans Secrets Manager, vous devez les fournir en texte brut, en utilisant le type Autre type de secrets. Pour de plus amples informations, veuillez consulter Utilisation de secrets pour accéder aux points de terminaison AWS Database Migration Service.

L'exemple de code suivant montre comment stocker des secrets de base de données en texte brut.

```
{
  "username": "dbuser",
  "password": "dbpassword"
}
```

Utilisation d'une base de données Amazon DocumentDB comme source pour des migrations de données homogènes dans AWS DMS

Cette section décrit comment configurer vos instances de base de données Amazon DocumentDB afin de les utiliser comme source pour des migrations de données homogènes.

Utilisez le nom d'utilisateur principal de l'instance Amazon DocumentDB comme compte utilisateur du fournisseur de données source compatible MongoDB pour des migrations de données homogènes vers. AWS DMS Le compte d'utilisateur principal a les rôles nécessaires qui lui permettent de configurer la capture des données modifiées. Si vous utilisez un compte autre que le compte utilisateur principal, le compte doit avoir le rôle root. Pour plus d'informations sur la création d'un utilisateur en tant que compte root, consultezDéfinition des autorisations pour utiliser Amazon DocumentDB en tant que source.

Pour activer la réplication logique, définissez le `change_stream_log_retention_duration` paramètre de votre groupe de paramètres de base de données sur un paramètre adapté à votre charge de travail transactionnelle. La modification de ce paramètre statique nécessite que vous redémarriez votre instance de base de données pour prendre effet. Avant de commencer la migration des données pour tous les types de tâches, y compris le chargement complet uniquement, activez les flux de modification Amazon DocumentDB pour toutes les collections d'une base de données donnée, ou uniquement pour certaines collections. Pour plus d'informations sur l'activation des flux de modifications pour Amazon DocumentDB, consultez la section Activation des flux de modifications dans le guide du développeur Amazon DocumentDB.

> **Note**
>
> AWS DMS utilise le flux de modifications Amazon DocumentDB pour capturer les modifications lors de la réplication en cours. Si Amazon DocumentDB supprime les enregistrements du flux de modifications avant que DMS ne les lise, vos tâches échoueront. Nous vous recommandons de définir le `change_stream_log_retention_duration` paramètre pour conserver les modifications pendant au moins 24 heures.

Pour utiliser Amazon DocumentDB pour une migration homogène des données, stockez les informations d'identification utilisateur dans Secrets Manager sous Identifiants pour la base de données Amazon DocumentDB.

Fonctionnalités d'utilisation d'une base de données compatible MongoDB comme source pour des migrations de données homogènes

- Vous pouvez migrer tous les index secondaires pris en charge par Amazon DocumentDB pendant la phase de chargement complet.
- AWS DMS fait migrer les collections en parallèle. Les migrations de données homogènes calculent les segments au moment de l'exécution en fonction de la taille moyenne de chaque document de la collection pour des performances optimales.
- DMS peut répliquer les index secondaires que vous créez lors de la phase CDC. DMS prend en charge cette fonctionnalité dans la version 6.0 de MongoDB.
- DMS prend en charge les documents dont le niveau d'imbrication est supérieur à 97.

Limitations liées à l'utilisation d'une base de données compatible avec MongoDB comme source pour des migrations de données homogènes

- Les documents ne peuvent pas avoir de nom de champ avec un $ préfixe.
- AWS DMS ne prend pas en charge la migration des collections de séries chronologiques.
- AWS DMS ne prend pas en charge `create drop` les événements `rename collection` DDL ou DDL pendant la phase CDC.
- AWS DMS ne prend pas en charge les types de données incohérents dans la collection pour le champ. `_id` Par exemple, la collection non prise en charge suivante comporte plusieurs types de données pour le `_id` champ.

```
rs0 [direct: primary] test> db.collection1.aggregate([
...   {
...     $group: {
...       _id: { $type: "$_id" },
...       count: { $sum: 1 }
...     }
...   }
... ])
[ { _id: 'string', count: 6136 }, { _id: 'objectId', count: 848033 } ]
```

- Pour les tâches CDC uniquement, AWS DMS seul le mode de `immediate` démarrage est pris en charge.
- AWS DMS ne prend pas en charge les documents contenant des caractères UTF8 non valides.

- AWS DMS ne prend pas en charge les collections fragmentées.

Bonnes pratiques d'utilisation d'une base de données compatible avec MongoDB comme source pour des migrations de données homogènes

- Pour plusieurs bases de données et collections volumineuses hébergées sur la même instance MongoDB, nous vous recommandons d'utiliser des règles de sélection pour chaque base de données et collection afin de répartir la tâche entre plusieurs tâches et projets de migration de données. Vous pouvez ajuster votre base de données et vos divisions de collecte pour optimiser les performances.

Création de fournisseurs de données cibles pour les migrations de données homogènes dans AWS DMS

Vous pouvez utiliser des bases de données compatibles avec MySQL, PostgreSQL et Amazon DocumentDB comme fournisseur de données cible pour des migrations de données homogènes dans. AWS DMS

Pour les versions de base de données prises en charge, consultezFournisseurs de données cibles pour les migrations de données homogènes DMS.

Votre fournisseur de données cible peut être une instance de base de données Amazon RDS ou un cluster de bases de données Amazon Aurora. Notez que la version de base de données de votre fournisseur de données cible doit être égale ou supérieure à la version de base de données de votre fournisseur de données source.

Rubriques

- Utilisation d'une base de données compatible MySQL en tant que cible pour les migrations de données homogènes dans AWS DMS
- Utilisation d'une base de données PostgreSQL en tant que cible pour les migrations de données homogènes dans AWS DMS
- Utilisation d'une base de données Amazon DocumentDB comme cible pour des migrations de données homogènes dans AWS DMS

Utilisation d'une base de données compatible MySQL en tant que cible pour les migrations de données homogènes dans AWS DMS

Vous pouvez utiliser une base de données compatible MySQL en tant que cible de migration pour les migrations de données homogènes dans AWS DMS.

AWS DMS nécessite certaines autorisations pour migrer les données vers la base de données Amazon RDS for MySQL, MariaDB ou Amazon Aurora MySQL cible. Utilisez le script suivant pour créer un utilisateur de base de données doté des autorisations requises dans la base de données cible MySQL.

```
CREATE USER 'your_user'@'%' IDENTIFIED BY 'your_password';

GRANT ALTER, CREATE, DROP, INDEX, INSERT, UPDATE, DELETE, SELECT, CREATE VIEW, CREATE
 ROUTINE, ALTER ROUTINE, EVENT, TRIGGER, EXECUTE, REFERENCES ON *.* TO 'your_user'@'%';
GRANT REPLICATION SLAVE, REPLICATION CLIENT  ON *.* TO 'your_user'@'%';
```

Dans l'exemple précédent, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

Utilisez le script suivant pour créer un utilisateur de base de données doté des autorisations requises dans la base de données MariaDB. Exécutez les requêtes GRANT pour toutes les bases de données que vous migrez vers AWS.

```
CREATE USER 'your_user'@'%' IDENTIFIED BY 'your_password';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE VIEW, CREATE
 ROUTINE, ALTER ROUTINE, EVENT, TRIGGER, EXECUTE,SLAVE MONITOR, REPLICATION SLAVE ON
 *.* TO 'your_user'@'%';
```

Dans l'exemple précédent, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

> **Note**
>
> Dans Amazon RDS, lorsque vous activez la sauvegarde automatique pour une instance de base de données MySQL/Maria, vous activez également la journalisation binaire. Lorsque ces paramètres sont activés, votre tâche de migration de données peut échouer avec l'erreur suivante lors de la création d'objets secondaires, tels que des fonctions, des procédures et des déclencheurs sur la base de données cible. Si la journalisation binaire est activée sur la

base de données cible, définissez `log_bin_trust_function_creators` sur `true` dans le groupe de paramètres de base de données avant de démarrer la tâche.

```
ERROR 1419 (HY000): You don't have the SUPER privilege and binary logging is
 enabled (you might want to use the less safe log_bin_trust_function_creators
 variable)
```

Limites d'utilisation d'une base de données compatible MySQL en tant que cible pour une migration de données homogène

Les limites suivantes s'appliquent à l'utilisation d'une base de données compatible MySQL en tant que cible pour une migration de données homogène :

- Le nom d'utilisateur que vous utilisez pour vous connecter à votre source de données présente les limites suivantes :
  - Il peut comporter de 2 à 64 caractères.
  - Il ne peut pas comporter d'espaces.
  - Il peut inclure les caractères suivants : a-z, A-Z, 0-9, trait de soulignement (_).
  - Il doit commencer par a-z ou A-Z.
- Le mot de passe que vous utilisez pour vous connecter à votre source de données présente les limites suivantes :
  - Il peut comporter de 1 à 128 caractères.
  - Il ne peut contenir aucun des éléments suivants : guillemet simple (’), guillemet double ("), point-virgule (;) ou espace.

Utilisation d'une base de données PostgreSQL en tant que cible pour les migrations de données homogènes dans AWS DMS

Vous pouvez utiliser une base de données PostgreSQL en tant que cible de migration pour les migrations de données homogènes dans AWS DMS.

AWS DMS nécessite certaines autorisations pour migrer les données vers la base de données Amazon RDS for PostgreSQL ou Amazon Aurora PostgreSQL cible. Utilisez le script suivant pour créer un utilisateur de base de données doté des autorisations requises dans la base de données cible PostgreSQL.

```
CREATE USER your_user WITH LOGIN PASSWORD 'your_password';
GRANT USAGE ON SCHEMA schema_name TO your_user;
GRANT CONNECT ON DATABASE db_name to your_user;
GRANT CREATE ON DATABASE db_name TO your_user;
GRANT CREATE ON SCHEMA schema_name TO your_user;
GRANT UPDATE, INSERT, SELECT, DELETE, TRUNCATE ON ALL TABLES IN SCHEMA schema_name
 TO your_user;
```

Dans l'exemple précédent, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

Pour activer la réplication logique pour votre cible RDS for PostgreSQL, définissez le paramètre `rds.logical_replication` de votre groupe de paramètres de base de données sur 1. Ce paramètre statique nécessite un redémarrage de l'instance de base de données ou du cluster de bases de données pour qu'il soit appliqué. Certains paramètres sont statiques et vous ne pouvez les définir qu'au démarrage du serveur. AWS DMS ignore les modifications apportées à leurs entrées dans le groupe de paramètres de base de données tant que vous ne redémarrez pas le serveur.

PostgreSQL utilise des déclencheurs pour implémenter des contraintes de clé étrangère. Pendant la phase de chargement complet, AWS DMS charge chaque table l'une après l'autre. Nous vous recommandons de désactiver les contraintes de clé étrangère sur la base de données cible pendant le chargement complet. Pour ce faire, utilisez l'une des méthodes suivantes.

- Désactivez temporairement tous les déclencheurs de votre instance et terminez le chargement complet.
- Modifiez la valeur du paramètre `session_replication_role` dans PostgreSQL.

  À un instant donné, un déclencheur peut être dans l'un des états suivants : `origin`, `replica`, `always` ou `disabled`. Lorsque vous définissez le paramètre `session_replication_role` sur `replica`, seuls les déclencheurs à l'état `replica` sont actifs. Sinon, les déclencheurs demeurent inactifs.

Limites d'utilisation d'une base de données compatible PostgreSQL en tant que cible pour une migration de données homogène

Les limites suivantes s'appliquent à l'utilisation d'une base de données compatible PostgreSQL en tant que cible pour une migration de données homogène :

- Le nom d'utilisateur que vous utilisez pour vous connecter à votre source de données présente les limites suivantes :
  - Il peut comporter de 2 à 64 caractères.
  - Il ne peut pas comporter d'espaces.
  - Il peut inclure les caractères suivants : a-z, A-Z, 0-9, trait de soulignement (_).
  - Il doit commencer par a-z ou A-Z.
- Le mot de passe que vous utilisez pour vous connecter à votre source de données présente les limites suivantes :
  - Il peut comporter de 1 à 128 caractères.
  - Il ne peut contenir aucun des éléments suivants : guillemet simple (’), guillemet double ("), point-virgule (;) ou espace.

Utilisation d'une base de données Amazon DocumentDB comme cible pour des migrations de données homogènes dans AWS DMS

Vous pouvez utiliser une base de données Amazon DocumentDB (compatible avec MongoDB) et un cluster Elastic DocumentDB comme cible de migration pour des migrations de données homogènes vers. AWS DMS

Pour exécuter des migrations de données homogènes pour une cible Amazon DocumentDB, vous pouvez créer soit un compte utilisateur doté de privilèges d'administrateur, soit un utilisateur disposant d'autorisations de lecture/écriture uniquement sur la base de données à migrer.

Les migrations de données homogènes prennent en charge tous les types de données BSON pris en charge par Amazon DocumentDB. Pour obtenir la liste de ces types de données, consultez la section Types de données du manuel Amazon DocumentDB Developer Guide.

Pour utiliser les fonctionnalités de partition du cluster Elastic DocumentDB pour migrer une collection non fragmentée depuis la source, créez une collection de partitions à migrer avant de démarrer la tâche de migration des données. Pour plus d'informations sur la collecte de partitions dans un cluster Amazon DocumentDB Elastic, consultez Étape 5 : Partagez votre collection dans le guide du développeur Amazon DocumentDB.

Pour une cible Amazon DocumentDB, AWS DMS prend en charge le mode `none` ou le mode `require` SSL.

Exécution de migrations de données homogènes dans AWS DMS

Vous pouvez l'utiliser Migrations de données homogènes AWS DMS pour migrer les données de votre base de données source vers le moteur équivalent sur Amazon Relational Database Service (Amazon RDS), Amazon Aurora ou Amazon DocumentDB. AWS DMSautomatise le processus de migration des données en utilisant des outils de base de données natifs dans vos bases de données source et cible.

Après avoir créé un profil d'instance et des fournisseurs de données compatibles pour les migrations de données homogènes, créez un projet de migration. Pour plus d'informations, consultez Création de projets de migration.

Les sections suivantes décrivent comment créer, configurer et exécuter des migrations de données homogènes.

Rubriques

- Création d'une migration de données dans AWS DMS
- Règles de sélection pour des migrations de données homogènes
- Gestion des migrations de données dans AWS DMS
- Surveillance des migrations de données dans AWS DMS
- Statuts des migrations de données homogènes dans AWS DMS
- Migration de données à partir de bases de données MySQL avec des migrations de données homogènes dans AWS DMS
- Migration de données à partir de bases de données PostgreSQL avec des migrations de données homogènes dans AWS DMS
- Migration de données depuis des bases de données MongoDB avec des migrations de données homogènes dans AWS DMS

Création d'une migration de données dans AWS DMS

Après avoir créé un projet de migration avec des fournisseurs de données compatibles du même type, vous pouvez utiliser ce projet pour les migrations de données homogènes. Pour plus d'informations, consultez Création de projets de migration.

Pour commencer à utiliser des migrations de données homogènes, créez une nouvelle migration de données. Vous pouvez créer plusieurs migrations de données homogènes de différents types dans un même projet de migration.

AWS DMS a le nombre maximal de migrations de données homogènes que vous pouvez créer pour votre Compte AWS. Pour en savoir plus sur les quotas de service AWS DMS, consultez Quotas pour AWS Database Migration Service.

Avant de créer une migration de données, assurez-vous de configurer les ressources requises, telles que vos bases de données sources et cibles, une politique et un rôle IAM, un profil d'instance et des fournisseurs de données. Pour plus d'informations, consultez Création de ressources IAM, Création de profils d'instance et Création de fournisseurs de données.

De plus, nous vous recommandons de ne pas utiliser de migrations de données homogènes pour migrer des données d'une version de base de données ultérieure vers une version de base de données antérieure. Vérifiez les versions des bases de données que vous utilisez pour les fournisseurs de données sources et cibles, puis mettez à niveau la version de la base de données cible, si nécessaire.

Pour créer une migration de données

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration, puis dans l'onglet Migrations de données, choisissez Créer une migration de données.
4. Pour Nom, entrez un nom pour votre migration de données. Assurez-vous d'utiliser un nom unique pour votre migration de données afin de pouvoir l'identifier facilement.
5. Pour Type de réplication, choisissez le type de migration de données que vous souhaitez configurer. Vous pouvez choisir l'une des options suivantes.

   - Chargement complet : migre vos données sources existantes.
   - Capture des données de chargement complet et de modification (CDC) : migre vos données sources existantes et réplique les modifications continues.
   - Capture de données modifiées (CDC) : réplique les modifications continues.

6. Cochez la case Activer les CloudWatch journaux pour stocker les journaux de migration des données sur Amazon CloudWatch. Si vous ne cochez pas cette case, vous ne pourrez pas voir les fichiers journaux lorsque la migration des données échoue.
7. (Facultatif) Développez Advanced settings (Paramètres avancés). Pour Nombre de tâches, entrez le nombre de threads parallèles qu'AWS DMS peut utiliser pour migrer vos données sources vers la cible.
8. Dans Fonction du service IAM, choisissez le rôle IAM que vous avez créé dans les prérequis. Pour plus d'informations, consultez Création d'un rôle IAM pour les migrations de données homogènes dans AWS DMS.
9. Configurez le Mode de démarrage pour les migrations de données de type Capture de données modifiées (CDC). Vous pouvez choisir l'une des options suivantes.

   - Immédiatement : démarre la réplication continue lorsque vous commencez la migration des données.
   - Utiliser un point de départ natif : démarre la réplication continue à partir du point spécifié.

     Pour les bases de données PostgreSQL, entrez le nom de l'emplacement de réplication logique dans Nom du slot et entrez le numéro de séquence du journal des transactions dans Point de départ natif.

     Pour les bases de données MySQL, entrez le numéro de séquence du journal des transactions dans Numéro de séquence de journal (LSN).
10. Configurez le Mode arrêt pour les migrations de données de type Capture de données modifiées (CDC) ou Capture des données de chargement complet et de modification (CDC). Vous pouvez choisir l'une des options suivantes.

    - N'arrêtez pas le CDC : AWS DMS poursuit la réplication continue jusqu'à ce que vous arrêtiez la migration de vos données.
    - Utilisation d'un point de temps du serveur : AWS DMS arrête la réplication continue à l'heure spécifiée.

      Si vous choisissez cette option, entrez la date et l'heure auxquelles vous souhaitez arrêter automatiquement la réplication continue dans Date et heure d'arrêt.
11. Choisissez Créer une migration de données.

AWS DMS crée votre migration de données et l'ajoute à la liste de l'onglet Migrations de données de votre projet de migration. Vous pouvez voir ici le statut de votre migration de données. Pour plus d'informations, consultez Statuts de migration.

> **Important**
>
> Pour les migrations de données de type Chargement complet et Capture des données de chargement complet et de modification (CDC), AWS DMS supprime toutes les données, tables et tous les autres objets de base de données de la base de données cible. Assurez-vous de disposer d'une sauvegarde de la base de données cible.

Une fois qu'AWS DMS a créé votre migration de données, le statut de celle-ci est défini sur Prêt. Pour migrer vos données, vous devez démarrer la migration des données manuellement. Pour ce faire, choisissez votre migration de données dans la liste. Ensuite, pour Actions, choisissez Démarrer. Pour plus d'informations, consultez Gestion des migrations de données.

Le premier lancement d'une migration de données homogène nécessite une certaine configuration. AWS DMS crée un environnement sans serveur pour la migration de vos données. Ce processus prend jusqu'à 15 minutes. Une fois que vous avez arrêté et redémarré votre migration de données, AWS DMS ne recrée plus l'environnement et vous pouvez accéder à votre migration de données plus rapidement.

Règles de sélection pour des migrations de données homogènes

Vous pouvez utiliser les règles de sélection pour choisir le schéma, les tables, ou les deux, que vous souhaitez inclure dans votre réplication.

> **Note**
>
> AWS DMSprend uniquement en charge les règles de sélection pour des migrations de données homogènes lorsque vous utilisez une base de données compatible MongoDB comme source.

Lorsque vous créez une tâche de migration de données, choisissez Ajouter une règle de sélection.

Pour les paramètres des règles, indiquez les valeurs suivantes :

- Schéma : choisissez Entrer un schéma.
- Nom du schéma : indiquez le nom du schéma que vous souhaitez répliquer ou utiliser `%` comme caractère générique.
- Nom de la table : Indiquez le nom de la table que vous souhaitez répliquer ou utiliser `%` comme caractère générique.

Par défaut, la seule action de règle prise en charge par DMS est`Include`, et le seul caractère générique pris en charge par DMS est. `%`

Example Migrer toutes les tables dans un schéma

L'exemple suivant migre toutes les tables d'un schéma nommé `dmsst` de la source vers votre point de terminaison cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-action": "include",
            "object-locator": {
                "schema-name": "dmsst",
                "table-name": "%"
            },
            "filters": [],
            "rule-id": "1",
            "rule-name": "1"
        }
    ]
}
```

Example Migrer certaines tables d'un schéma

L'exemple suivant fait migrer toutes les tables dont le nom commence par`collectionTest`, d'un schéma nommé `dmsst` dans votre source vers votre point de terminaison cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-action": "include",
```

```
            "object-locator": {
                "schema-name": "dmsst",
                "table-name": "collectionTest%"
            },
            "filters": [],
            "rule-id": "1",
            "rule-name": "1"
        }
    ]
}
```

Example Migrer des tables spécifiques à partir de plusieurs schémas

L'exemple suivant fait migrer certaines tables de plusieurs schémas nommés `dmsst` et figurant `Test` dans votre source vers votre point de terminaison cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-action": "include",
            "object-locator": {
                "schema-name": "dmsst",
                "table-name": "collectionTest1"
            },
            "filters": [],
            "rule-id": "1",
            "rule-name": "1"
        },
        {
            "rule-type": "selection",
            "rule-action": "include",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "products"
            },
            "filters": [],
            "rule-id": "2",
            "rule-name": "2"
        }
    ]
}
```

Gestion des migrations de données dans AWS DMS

Une fois que vous avez créé une migration de données, AWS DMS ne la démarre pas automatiquement. Vous démarrez une migration de données manuellement, si nécessaire.

Avant de démarrer une migration de données, vous pouvez modifier tous ses paramètres. Après avoir démarré votre migration de données, vous ne pouvez pas modifier le type de réplication. Pour utiliser un autre type de réplication, créez une nouvelle migration de données.

Pour démarrer une migration de données

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration. Dans l'onglet Migrations de données, choisissez votre migration de données. La page Récapitulatif de votre migration de données s'ouvre.
4. Pour Actions, choisissez Start (Démarrer).

   Ensuite, AWS DMS crée un environnement sans serveur pour votre migration de données. Ce processus prend jusqu'à 15 minutes.

Une fois que vous avez démarré une migration de données, AWS DMS définit son statut sur Démarrage en cours. Le statut suivant utilisé par AWS DMS pour votre migration de données dépend du type de réplication que vous choisissez dans les paramètres de migration des données. Pour plus d'informations, consultez Statuts de migration.

Pour modifier une migration de données

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration. Dans l'onglet Migrations de données, choisissez votre migration de données. La page Récapitulatif de votre migration de données s'ouvre.
4. Sélectionnez Modifier.
5. Configurez les paramètres de votre migration de données.

> **Important**
>
> Si vous avez démarré une migration de données, vous ne pouvez pas modifier le type de réplication.

6. Pour consulter vos journaux de migration de données sur Amazon CloudWatch, cochez la case Activer les CloudWatch journaux.
7. Sélectionnez Enregistrer les modifications.

Une fois qu'AWS DMS a démarré une migration de données, vous pouvez l'arrêter. Pour ce faire, choisissez votre migration de données dans l'onglet Migrations de données. Ensuite, pour Actions, choisissez Arrêter.

Une fois que vous avez arrêté une migration de données, AWS DMS définit son statut sur Arrêt. Ensuite, AWS DMS définit le statut de cette migration de données sur Arrêté. Une fois qu'AWS DMS a arrêté une migration de données, vous pouvez modifier, reprendre, redémarrer ou supprimer votre migration de données.

Pour poursuivre la réplication des données, choisissez la migration de données que vous avez arrêtée dans l'onglet Migrations de données. Ensuite, pour Actions, choisissez Reprise du traitement.

Pour redémarrer le chargement des données, choisissez la migration de données que vous avez arrêtée dans l'onglet Migrations de données. Ensuite, pour Actions, choisissez Redémarrer. AWS DMS supprime toutes les données de la base de données cible et démarre la migration des données à partir de zéro.

Vous pouvez supprimer une migration de données que vous avez arrêtée ou que vous n'avez pas démarrée. Pour supprimer une migration de données, choisissez-la dans l'onglet Migrations de données. Ensuite, pour Actions, choisissez Supprimer. Pour supprimer votre projet de migration, arrêtez et supprimez toutes les migrations de données.

Surveillance des migrations de données dans AWS DMS

Une fois que vous avez démarré votre migration de données homogène, vous pouvez surveiller son statut et sa progression. Les migrations de jeux de données de grande taille, des centaines de gigaoctets par exemple, prennent des heures. Pour gérer la fiabilité, la disponibilité et les performances de votre migration de données, surveillez régulièrement sa progression.

Pour vérifier le statut et la progression de votre migration de données

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration et accédez à l'onglet Migrations de données.
4. Pour votre migration de données, consultez la colonne Statut. Pour plus d'informations sur les valeurs de cette colonne, consultez Statuts de migration.
5. Pour une migration de données en cours, la colonne Progrès de la migration affiche le pourcentage de données migrées.

Pour vérifier les détails de votre migration de données

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration. Dans l'onglet Migrations de données, choisissez votre migration de données.
4. L'onglet Détails affiche la progression de la migration. Vous pouvez notamment voir les métriques suivantes.

   - Adresse IP publique : adresse IP publique de votre migration de données. Vous avez besoin de cette valeur pour configurer un réseau. Pour plus d'informations, consultez Configuration d'un réseau.
   - Tables chargées : nombre de tables chargées avec succès.
   - Tables en cours de chargement : nombre de tables en cours de chargement.
   - Tables mises en attente : nombre de tables en attente de chargement.
   - Tables erronées : nombre de tables dont le chargement a échoué.
   - Temps écoulé : temps écoulé depuis le début de la migration de données.
   - Latence du CDC : temps moyen qui s'écoule entre le moment où une modification est effectuée sur une table source et le moment où AWS DMS applique cette modification à la table cible.
   - Migration démarrée : heure à laquelle vous avez démarré cette migration de données.
   - Migration arrêtée : heure à laquelle vous avez arrêté cette migration de données.

5. Pour afficher les fichiers journaux relatifs à votre migration de données, choisissez Afficher CloudWatch les journaux sous Paramètres de migration de données homogènes. Vous pouvez activer les CloudWatch journaux lorsque vous créez ou modifiez une migration de données. Pour plus d'informations, consultez Création d'une migration de données et Gestion des migrations de données.

Vous pouvez utiliser les CloudWatch alarmes ou les événements Amazon pour suivre de près la migration de vos données. Pour plus d'informations, consultez Que sont Amazon CloudWatch, Amazon CloudWatch Events et Amazon CloudWatch Logs ? dans le guide de CloudWatch l'utilisateur Amazon. Notez que l'utilisation d'Amazon est payante CloudWatch.

Pour des migrations de données homogènes, AWS DMS inclut les métriques suivantes dans Amazon CloudWatch.

Métrique	Description
OverallCDCLatency	Latence globale pendant la phase de CDC. Pour les bases de données MySQL, cette métrique indique le nombre de secondes qui s'écoulent entre la modification dans le journal binaire source et la réplication de cette modification. Pour les bases de données PostgreSQL, cette métrique indique le nombre de secondes qui s'écoulent entre `last_msg_receipt_time` et `last_msg_send_time` à partir de la vue `pg_stat_subscription` . Unités : secondes
StorageConsumption	Espace de stockage consommé par votre migration de données. Unités : octets

Statuts des migrations de données homogènes dans AWS DMS

Pour chaque migration de données que vous exécutez, AWS DMS affiche son Statut dans la console AWS DMS. La liste suivante répertorie les statuts disponibles.

- `Creating` : AWS DMS crée la migration de données.
- `Ready` : la migration de données est prête à démarrer.
- `Starting` : AWS DMS crée l'environnement sans serveur pour votre migration de données. Ce processus prend jusqu'à 15 minutes.
- `Load running` : AWS DMS effectue la migration de chargement complet.
- `Load complete, replication ongoing` : AWS DMS a terminé le chargement complet et réplique maintenant les modifications continues. AWS DMS utilise ce statut uniquement pour les migrations de données de type Chargement complet + capture des données de modification (CDC).
- `Replication ongoing` : AWS DMS réplique les modifications continues. AWS DMS utilise ce statut uniquement pour les migrations de type Capture des données de modification (CDC).
- `Reloading target` : AWS DMS redémarre une migration de données et exécute le type de migration spécifié.
- `Stopping` : AWS DMS arrête la migration des données. AWS DMS définit ce statut une fois que vous avez choisi d'arrêter la migration des données dans le menu Actions.
- `Stopped` : AWS DMS a arrêté la migration des données.
- `Failed` : la migration de données a échoué. Pour plus d'informations, consultez les fichiers journaux.

  Pour afficher les fichiers journaux, choisissez votre migration de données dans l'onglet Migrations de données. Choisissez ensuite Afficher les CloudWatch journaux sous Paramètres de migration de données homogènes.

  > **Important**
  >
  > Vous pouvez consulter les fichiers journaux si vous cochez la case Activer les CloudWatch journaux lorsque vous créez votre migration de données.

- `Deleting` : AWS DMS supprime la migration de données. AWS DMS définit ce statut une fois que vous avez choisi de supprimer la migration des données dans le menu Actions.

Migration de données à partir de bases de données MySQL avec des migrations de données homogènes dans AWS DMS

Vous pouvez utiliser Migrations de données homogènes pour migrer une base de données MySQL autogérée vers RDS for MySQL ou Aurora MySQL. AWS DMS crée un environnement sans serveur pour la migration de vos données. Pour différents types de migrations de données, AWS DMS utilise différents outils de base de données MySQL natifs.

Pour des migrations de données homogènes de type Chargement complet, AWS DMS utilise mydumper pour lire les données à partir de la base de données source et les stocker sur le disque connecté à l'environnement sans serveur. Une fois qu'AWS DMS a lu toutes vos données sources, il utilise myloader dans la base de données cible pour restaurer vos données.

Pour des migrations de données homogènes de type Capture des données de chargement complet et de modification (CDC), AWS DMS utilise mydumper pour lire les données à partir de la base de données source et les stocker sur le disque connecté à l'environnement sans serveur. Une fois qu'AWS DMS a lu toutes vos données sources, il utilise myloader dans la base de données cible pour restaurer vos données. Une fois qu'AWS DMS a terminé le chargement complet, il configure la réplication du journal binaire avec la position du journal binaire définie au début du chargement complet. Pour éviter toute incohérence des données, définissez Nombre de tâches sur 1 afin de capturer l'état cohérent des données existantes. Pour plus d'informations, consultez Création d'une migration de données.

Pour des migrations de données homogènes de type Capture de données modifiées (CDC), AWS DMS exige que le Point de départ CDC natif soit spécifié pour démarrer la réplication. Si vous indiquez le point de départ natif de la CDC, AWS DMS capture les modifications à partir de ce point. Vous pouvez également choisir Immédiatement dans les paramètres de migration des données pour capturer automatiquement le point de départ de la réplication lorsque la migration des données commence réellement.

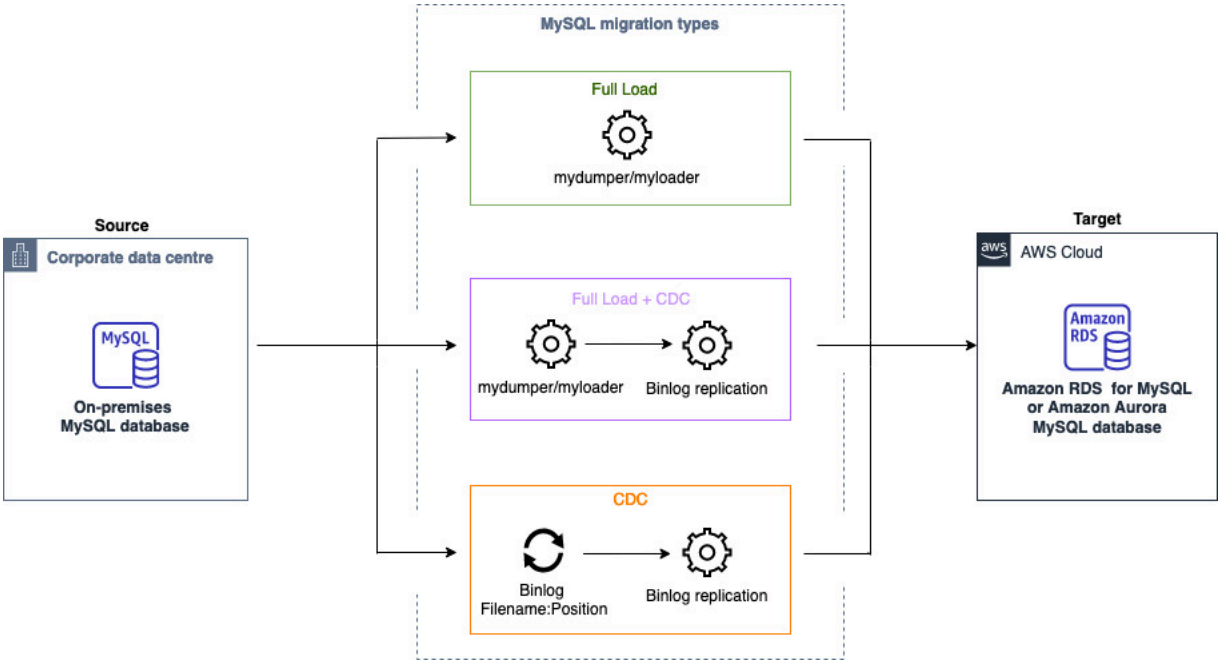
> **Note**
>
> Pour qu'une migration de CDC uniquement fonctionne correctement, tous les schémas et objets de base de données source doivent déjà être présents dans la base de données cible. La cible peut toutefois contenir des objets qui ne sont pas présents sur la source.

Vous pouvez utiliser l'exemple de code suivant pour obtenir le numéro de séquence de journal (LSN) actuel dans la base de données MySQL.

```
show master status
```

Cette requête renvoie le nom du fichier journal binaire et la position. Pour le point de départ natif, utilisez une combinaison du nom du fichier journal binaire et de la position. Par exemple, `mysql-bin-changelog.000024:373`. Dans cet exemple, `mysql-bin-changelog.000024` est le nom du fichier journal binaire et `373` est la position où AWS DMS démarre la capture des modifications.

Le diagramme suivant montre le processus d'utilisation des migrations de données homogènes dans AWS DMS pour migrer une base de données MySQL vers RDS for MySQL ou Aurora MySQL.



Migration de données à partir de bases de données PostgreSQL avec des migrations de données homogènes dans AWS DMS

Vous pouvez utiliser Migrations de données homogènes pour migrer une base de données PostgreSQL autogérée vers RDS for PostgreSQL ou Aurora PostgreSQL. AWS DMS crée un environnement sans serveur pour la migration de vos données. Pour différents types de migrations de données, AWS DMS utilise différents outils de base de données PostgreSQL natifs.

Pour des migrations de données homogènes de type Chargement complet, AWS DMS utilise pg_dump pour lire les données à partir de la base de données source et les stocker sur le disque

connecté à l'environnement sans serveur. Une fois qu'AWS DMS a lu toutes vos données sources, il utilise pg_restore dans la base de données cible pour restaurer vos données.

Pour des migrations de données homogènes de type Capture des données de chargement complet et de modification (CDC), AWS DMS utilise pg_dump pour lire les données à partir de la base de données source et les stocker sur le disque connecté à l'environnement sans serveur. Une fois qu'AWS DMS a lu toutes vos données sources, il utilise pg_restore dans la base de données cible pour restaurer vos données. Une fois qu'AWS DMS a terminé le chargement complet, il bascule automatiquement vers un modèle de diffuseur de publication et d'abonné pour la réplication logique. Dans ce modèle, un ou plusieurs abonnés s'abonnent à une ou plusieurs publications sur un nœud de diffuseur de publication.

Pour des migrations de données homogènes de type Capture de données modifiées (CDC), AWS DMS exige que le point de départ natif soit spécifié pour démarrer la réplication. Si vous indiquez le point de départ natif, AWS DMS capture les modifications à partir de ce point. Vous pouvez également choisir Immédiatement dans les paramètres de migration des données pour capturer automatiquement le point de départ de la réplication lorsque la migration des données commence réellement.

> **Note**
>
> Pour qu'une migration de CDC uniquement fonctionne correctement, tous les schémas et objets de base de données source doivent déjà être présents dans la base de données cible. La cible peut toutefois contenir des objets qui ne sont pas présents sur la source.

Vous pouvez utiliser l'exemple de code suivant pour obtenir le point de départ natif dans la base de données PostgreSQL.

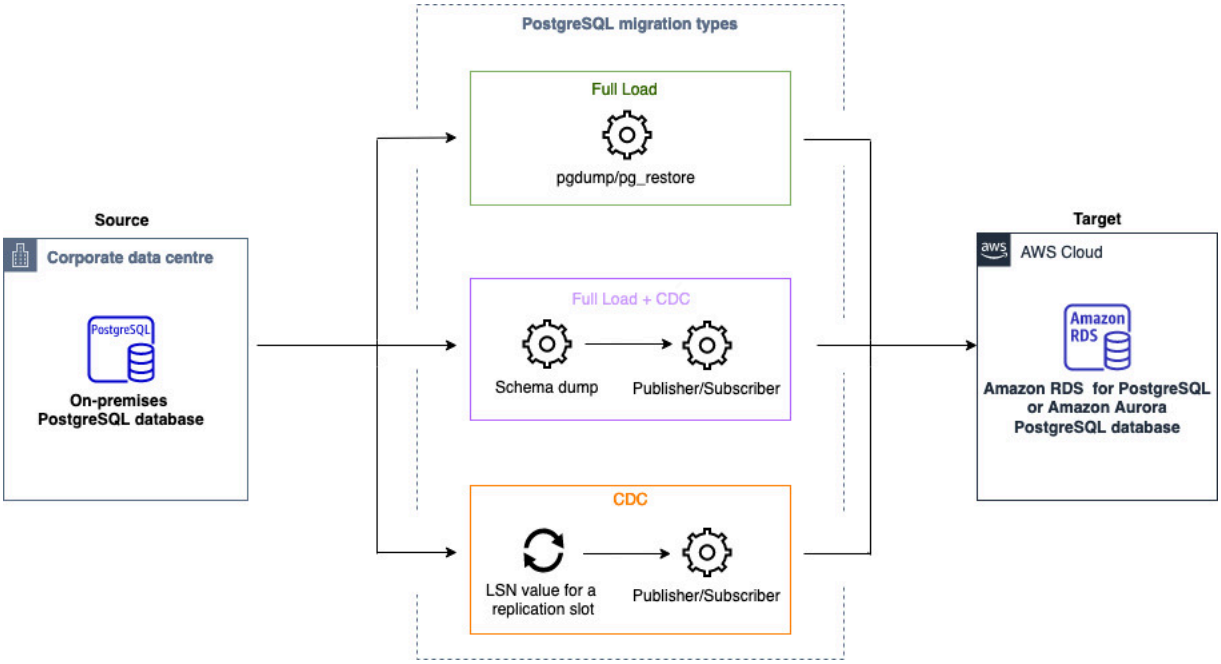
```
select confirmed_flush_lsn from pg_replication_slots where
 slot_name='migrate_to_target';
```

Cette requête utilise la vue `pg_replication_slots` de la base de données PostgreSQL pour capturer la valeur du numéro de séquence de journal (LSN).

Une fois qu'AWS DMS a défini le statut de votre migration de données homogène PostgreSQL sur Arrêté, Échec ou Supprimé, de diffuseur de publication et la réplication ne sont pas supprimés. Si vous ne souhaitez pas reprendre la migration, supprimez l'emplacement de réplication et le diffuseur de publication à l'aide de la commande suivante.

```
SELECT pg_drop_replication_slot('migration_subscriber_{ARN}');
            DROP PUBLICATION publication_{ARN};
```

Le diagramme suivant montre le processus d'utilisation des migrations de données homogènes dans AWS DMS pour migrer une base de données PostgreSQL vers RDS for PostgreSQL ou Aurora PostgreSQL.



Migration de données depuis des bases de données MongoDB avec des migrations de données homogènes dans AWS DMS

Vous pouvez l'utiliser Migrations de données homogènes pour migrer une base de données MongoDB autogérée vers Amazon DocumentDB. AWS DMScrée un environnement sans serveur pour la migration de vos données. Pour différents types de migrations de données, AWS DMS utilise différents outils de base de données MongoDB natifs.

Pour des migrations de données homogènes du type `mongodump` à chargement complet, AWS DMS il permet de lire les données de votre base de données source et de les stocker sur le disque connecté à l'environnement sans serveur. Après avoir AWS DMS lu toutes vos données sources, il les utilise `mongorestore` dans la base de données cible pour restaurer vos données.

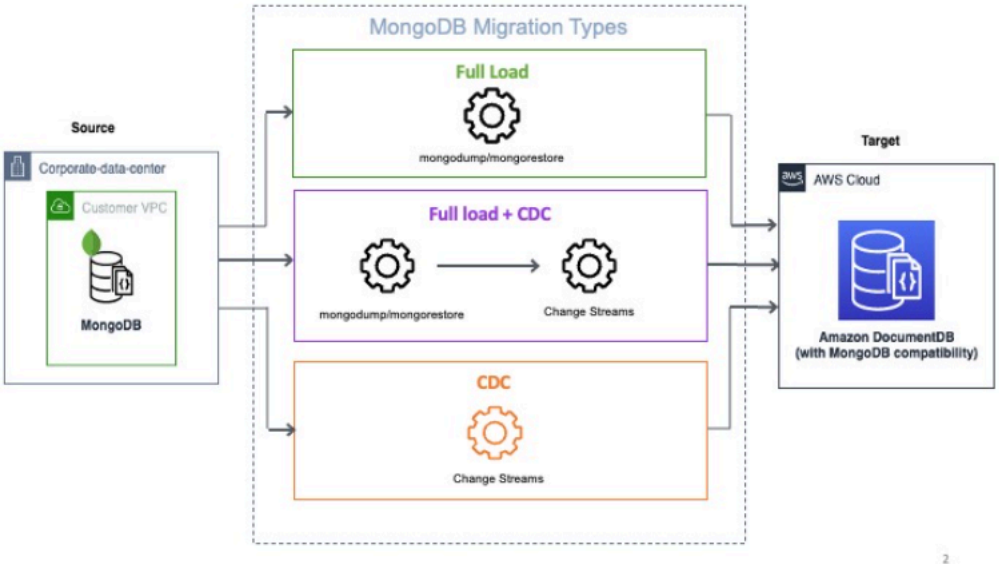
Pour des migrations de données homogènes du type « Full load and change data capture » (CDC), il AWS DMS permet `mongodump` de lire les données de votre base de données source et de les stocker sur le disque connecté à l'environnement sans serveur. Après avoir AWS DMS lu toutes vos

données sources, il les utilise `mongorestore` dans la base de données cible pour restaurer vos données. Une fois qu'AWS DMS a terminé le chargement complet, il bascule automatiquement vers un modèle de diffuseur de publication et d'abonné pour la réplication logique. Dans ce modèle, nous recommandons de dimensionner le journal afin de conserver les modifications pendant au moins 24 heures.

Pour des migrations de données homogènes du type Change data capture (CDC), choisissez `immediately` dans les paramètres de migration des données de capturer automatiquement le point de départ de la réplication lorsque la migration des données commence réellement.

> **Note**
>
> Pour toute collection nouvelle ou renommée, vous devez créer une nouvelle tâche de migration de données pour ces collections sous forme de migrations de données homogènes. Pour une source compatible avec MongoDB, AWS DMS ne prend pas en charge `create` les opérations. `rename` `drop collection`

Le schéma suivant montre le processus d'utilisation de migrations de données homogènes AWS DMS pour migrer une base de données MongoDB vers Amazon DocumentDB.



Résolution des problèmes liés aux migrations de données homogènes dans AWS DMS

La liste suivante répertorie les mesures à prendre lorsque vous rencontrez des problèmes liés aux migrations de données homogènes dans AWS DMS.

Rubriques

- Je ne parviens pas à créer une migration de données homogène dans AWS DMS
- Je ne parviens pas à démarrer une migration de données homogène dans AWS DMS
- Je ne parviens pas à me connecter à la base de données cible lors de l'exécution d'une migration de données dans AWS DMS
- AWS DMS migre les vues en tant que tables dans PostgreSQL

Je ne parviens pas à créer une migration de données homogène dans AWS DMS

Si vous recevez un message d'erreur indiquant qu'AWS DMS ne parvient pas à se connecter à vos fournisseurs de données après avoir choisi Créer une migration de données, assurez-vous d'avoir configuré le rôle IAM requis. Pour de plus amples informations, veuillez consulter Création d'un rôle IAM.

Si vous avez configuré le rôle IAM et que vous recevez toujours ce message d'erreur, ajoutez ce rôle IAM à votre utilisateur de clé dans la configuration de clé AWS KMS. Pour plus d'informations, consultez Autorise les utilisateurs de clés à utiliser la clé KMS dans le Manuel du développeur AWS Key Management Service.

Je ne parviens pas à démarrer une migration de données homogène dans AWS DMS

Si vous obtenez le statut `Failed` lorsque vous démarrez une migration de données dans votre projet de migration, vérifiez les versions de vos fournisseurs de données sources et cibles. Pour ce faire, exécutez la requête `SELECT VERSION();` dans la base de données MySQL ou PostgreSQL. Veillez à utiliser la version de base de données prise en charge.

Pour obtenir la liste des bases de données sources prises en charge, consultez Sources pour les migrations de données homogènes DMS.

Pour obtenir la liste des bases de données cibles prises en charge, consultez Cibles pour les migrations de données homogènes DMS.

Si vous utilisez une version de base de données non prise en charge, mettez à niveau la base de données source ou cible, puis réessayez.

Vérifiez le message d'erreur relatif à votre migration de vos données dans la console AWS DMS. Pour ce faire, ouvrez votre projet de migration et choisissez votre migration de données. Dans l'onglet Détails, vérifiez le Dernier message d'échec sous Général.

Enfin, analysez le journal CloudWatch. Pour ce faire, ouvrez votre projet de migration et choisissez votre migration de données. Dans l'onglet Détails, choisissez Afficher les journaux CloudWatch.

Je ne parviens pas à me connecter à la base de données cible lors de l'exécution d'une migration de données dans AWS DMS

Si le message d'erreur Impossible de se connecter à la cible s'affiche, prenez les mesures suivantes.

1. Assurez-vous que le groupe de sécurité attaché à vos bases de données sources et cibles contient une règle pour le trafic entrant et sortant. Pour de plus amples informations, veuillez consulter Configuration de la réplication continue des données.
2. Vérifiez la liste de contrôle d'accès (ACL) du réseau et les règles de table de routage.
3. Votre base de données doit être accessible à partir du VPC que vous avez créé. Ajoutez des adresses IP publiques dans les groupes de sécurité de VPC et autorisez les connexions d'entrée dans votre pare-feu.
4. Dans l'onglet Migrations de données de votre projet de migration, choisissez votre migration de données. Notez l'adresse IP publique sous Connectivité et sécurité dans l'onglet Détails. Ensuite, autorisez l'accès à partir de l'adresse IP publique de votre migration de données dans vos bases de données sources et cibles.
5. Pour une réplication continue des données, assurez-vous que vos bases de données sources et cibles peuvent communiquer entre elles.

Pour plus d'informations, veuillez consulter la rubrique Contrôler le trafic vers les ressources à l'aide de groupes de sécurité dans le Guide de l'utilisateur Amazon Virtual Private Cloud.

AWS DMS migre les vues en tant que tables dans PostgreSQL

La migration de données homogène ne prend pas en charge la migration des vues en tant que vues dans PostgreSQL. Pour PostgreSQL, AWS DMS migre les vues en tant que tables.

Travailler avec des fournisseurs de données, des profils d'instance et des projets de migration dans AWS DMS

Lorsque vous utilisez la conversion de schéma DMS et des migrations de données homogènes dans AWS Database Migration Service, vous travaillez sur des projets de migration. Les projets de migration AWS DMS utilisent eux-mêmes des groupes de sous-réseaux, des profils d’instance et des fournisseurs de données.

Un sous-réseau est une plage d'adresses IP dans votre VPC. Un groupe de sous-réseaux de réplication inclut des sous-réseaux provenant de différentes zones de disponibilité que votre profil d'instance peut utiliser. Notez qu'un groupe de sous-réseaux de réplication est une ressource DMS et qu'il est distinct des groupes de sous-réseaux utilisés par Amazon VPC et Amazon RDS.

Un profil d’instance spécifie les paramètres réseau et de sécurité pour l’environnement sans serveur dans lequel s’exécute votre projet de migration.

Un fournisseur de données stocke un type de magasin de données et les informations d’emplacement de la base de données. Après avoir ajouté un fournisseur de données à votre projet de migration, vous devez fournir les informations d'identification de la base de données depuis AWS Secrets Manager. AWS DMS utilise ces informations pour se connecter à votre base de données.

Après avoir créé des fournisseurs de données, votre profil d'instance et d'autres AWS ressources, vous pouvez créer un projet de migration. Un projet de migration décrit le profil de votre instance, les fournisseurs de données source et cible, ainsi que les secrets de AWS Secrets Manager. Vous pouvez créer plusieurs projets de migration pour différents fournisseurs de données source et cible.

Vous effectuez la majeure partie de votre travail dans le projet de migration. Pour la conversion de schéma DMS, vous utilisez un projet de migration pour évaluer les objets de votre fournisseur de données source et les convertir dans un format compatible avec la base de données cible. Vous pouvez ensuite appliquer le code converti à votre fournisseur de données cible ou l’enregistrer en tant que script SQL. Pour des migrations de données homogènes, vous utilisez un projet de migration pour migrer les données de la base de données source vers une base de données cible du même type dans le AWS Cloud.

Les projets de migration se AWS DMS font uniquement sans serveur. AWS DMS provisionne automatiquement les ressources cloud pour vos projets de migration.

AWS DMS possède le nombre maximum de profils d'instance, de fournisseurs de données et de projets de migration que vous pouvez créer pour votre Compte AWS. Pour en savoir plus sur les quotas de service AWS DMS , consultez Quotas pour AWS Database Migration Service.

Rubriques

- Création d'un groupe de sous-réseaux pour un projet de AWS DMS migration
- Création de profils d'instance pour AWS Database Migration Service
- Création de fournisseurs de données dans AWS Database Migration Service
- Création de projets de migration dans AWS Database Migration Service
- Gestion des projets de migration dans AWS Database Migration Service

Création d'un groupe de sous-réseaux pour un projet de AWS DMS migration

Avant de créer un profil d'instance, configurez un groupe de sous-réseaux pour votre profil d'instance.

Pour créer un groupe de sous-réseaux

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Dans le volet de navigation, choisissez Groupes de sous-réseaux, puis Créer un groupe de sous-réseaux.
3. Dans Nom, entrez un nom unique pour votre groupe de sous-réseaux.
4. Dans Description, entrez une courte description pour votre groupe de sous-réseaux.
5. Pour VPC, choisissez un VPC qui possède au moins un sous-réseau dans au moins deux zones de disponibilité.
6. Pour Ajouter des sous-réseaux, choisissez les sous-réseaux à inclure dans le groupe de sous-réseaux. Vous devez choisir des sous-réseaux dans au moins deux zones de disponibilité.

   Pour vous connecter aux bases de données Amazon RDS, ajoutez des sous-réseaux publics dans votre groupe de sous-réseaux. Pour vous connecter aux bases de données sur site, ajoutez des sous-réseaux privés dans votre groupe de sous-réseaux.
7. Choisissez Créer groupe de sous-réseaux.

Création de profils d'instance pour AWS Database Migration Service

Vous pouvez créer plusieurs profils d'instance dans la AWS DMS console. Assurez-vous de sélectionner un profil d'instance à utiliser pour chaque projet de migration que vous créez dans AWS DMS.

Pour créer un profil d'instance

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Dans le volet de navigation, choisissez Profils d'instance.
3. Choisissez Créer un profil d'instance.
4. Sur la page Créer un profil d'instance, entrez une valeur descriptive pour le Nom de votre profil d'instance.
5. Pour Type de réseau, choisissez Mode double pile pour créer un profil d'instance prenant en charge l'adressage IPv4 et IPv6. Conservez l'option par défaut pour créer un profil d'instance qui prend uniquement en charge l'adressage IPv4.
6. Ensuite, choisissez Cloud privé virtuel (VPC) pour exécuter votre instance du type de réseau sélectionné. Choisissez ensuite un Groupe de sous-réseaux et des Groupes de sécurité VPC pour votre profil d'instance.

   Pour vous connecter aux bases de données Amazon RDS, utilisez un groupe de sous-réseaux qui inclut des sous-réseaux publics. Pour vous connecter aux bases de données sur site, utilisez un groupe de sous-réseaux qui inclut des sous-réseaux privés. Assurez-vous d'avoir configuré votre réseau de manière à AWS DMS pouvoir accéder à votre base de données source sur site à l'aide de l'adresse IP publique de la passerelle NAT. Pour plus d'informations, consultez Création d'un VPC basé sur Amazon VPC.
7. (Facultatif) Si vous créez un projet de migration pour la conversion de schéma DMS, pour Paramètres de conversion de schéma - facultatif, choisissez un compartiment Amazon S3 pour stocker les informations de votre projet de migration. Choisissez ensuite le rôle AWS Identity and Access Management (IAM) qui donne accès à ce compartiment Amazon S3. Pour plus d'informations, consultez Créer un compartiment Amazon S3.
8. Choisissez Créer un profil d'instance.

Une fois votre profil d'instance créé, vous pouvez le modifier ou le supprimer.

Pour modifier un profil d'instance

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Instance profiles (Profils d'instance). La page Profils d'instance s'ouvre.
3. Choisissez votre profil d'instance, puis Modifier.
4. Mettez à jour le nom de votre profil d'instance et modifiez les paramètres du VPC ou du compartiment Amazon S3.
5. Sélectionnez Enregistrer les modifications.

Pour supprimer un profil d'instance

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Instance profiles (Profils d'instance). La page Profils d'instance s'ouvre.
3. Choisissez votre profil d'instance, puis Supprimer.
4. Choisissez Supprimer pour confirmer votre choix.

Création de fournisseurs de données dans AWS Database Migration Service

Vous pouvez créer des fournisseurs de données et les utiliser dans des projets de AWS DMS migration. Votre fournisseur de données peut être un moteur autogéré s'exécutant sur site ou sur une instance Amazon EC2. De plus, votre fournisseur de données peut être un moteur entièrement géré tel que Amazon Relational Database Service (Amazon RDS) ou Amazon Aurora.

Pour chaque base de données, vous pouvez créer un fournisseur de données unique. Vous pouvez utiliser un même fournisseur de données dans plusieurs projets de migration.

Avant de créer un projet de migration, assurez-vous d'avoir créé au moins deux fournisseurs de données. L'un de vos fournisseurs de données doit être sur un Service AWS. Vous ne pouvez pas utiliser AWS DMS pour convertir vos schémas ou migrer vos données vers une base de données sur site.

La procédure suivante explique comment créer des fournisseurs de données dans l'assistant de AWS DMS console.

Pour créer un fournisseur de données

1. Connectez-vous à la AWS DMS console AWS Management Console, puis ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Fournisseurs de données. La page Fournisseurs de données s'ouvre.
3. Choisissez Créer un fournisseur de données. Le tableau suivant décrit les paramètres.

Option	Action
Configuration	Choisissez d'entrer manuellement les informations relatives à votre fournisseur de données ou d'utiliser l'instance de base de données Amazon RDS.
Nom	Entrez un nom pour votre fournisseur de données. Assurez-vous d'utiliser un nom unique pour votre fournisseur de données afin de pouvoir l'identifier facilement.
Engine type (Type de moteur)	Choisissez le type de moteur de base de données pour votre fournisseur de données.
Server name	Entrez le nom du service DNS ou l'adresse IP de votre serveur de base de données. Le nom du serveur du fournisseur de données utilisé pour une réplication homogène doit commencer par un caractère alphanumérique, et ne peut comporter que des caractères alphanumériques, des tirets (-), des points (.) et des traits de soulignement (_).
Port	Entrez le port utilisé pour vous connecter à votre serveur de base de données.
ID de service (SID) ou nom du service	Entrez l'ID du système Oracle (SID). Pour trouver le SID Oracle, soumettez la requête suivante à votre base de données Oracle :

Option	Action
	`SELECT sys_context('userenv','instance_name') AS SID FROM dual;`
Nom de base de données	Entrez le nom de la base de données pour ce fournisseur de données. Le nom de la base de données du fournisseur de données utilisée pour une réplication homogène peut comporter jusqu'à 63 caractères et ne doit pas contenir d'espaces.
Mode SSL (Secure Socket Layer)	Choisissez un mode SSL si vous souhaitez activer le chiffrement de connexion pour ce fournisseur de données. Selon le mode que vous sélectionnez, vous devrez peut-être fournir les informations concernant le certificat et le certificat du serveur. Pour plus de détails, consultez la section Utilisation du protocole SSL avec AWS Database Migration Service.
Mode d'authentification	Pour une source MongoDB, le mode d'authentification AWS DMS utilisé pour authentifier la connexion du point de terminaison.
Source d'authentification	Pour une source MongoDB, le nom de la base de données MongoDB à utiliser pour valider vos informations d'authentification.
Mécanisme d'authentification	Pour une source MongoDB, méthode d'authentification utilisée par MongoDB pour chiffrer le mot de passe.

4. Choisissez Créer un fournisseur de données.

Après avoir créé un fournisseur de données, assurez-vous d'ajouter les informations d'identification de connexion à la base de données dans AWS Secrets Manager.

Création de projets de migration dans AWS Database Migration Service

Avant de créer un projet de migration dans AWS DMS, assurez-vous de créer les ressources suivantes :

- Fournisseurs de données décrivant vos bases de données source et cible
- Secrets contenant les informations d'identification de base de données stockées dans AWS Secrets Manager
- Le rôle AWS Identity and Access Management (IAM) qui donne accès à Secrets Manager
- Un profil d'instance qui inclut les paramètres réseau et de sécurité

Pour créer un projet de migration

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez Créer un projet de migration. Le tableau suivant décrit les paramètres.

Option	Action
Nom	Entrez un nom pour votre projet de migration. Assurez-vous d'utiliser un nom unique pour votre projet de migration afin de pouvoir l'identifier facilement.
Profil d'instance	Choisissez le profil d'instance à utiliser pour votre projet de migration.
Source	Choisissez Parcourir, puis choisissez votre fournisseur de données source.
ID secret	Choisissez l'Amazon Resource Name (ARN) de votre secret dans Secrets Manager qui stocke les informations d'identif ication de la base de données source.

Option	Action
Rôle IAM	Choisissez un rôle IAM pour fournir l'accès aux informations d'identification de la base de données source dans Secrets Manager.
Cible	Choisissez Parcourir, puis choisissez votre fournisseur de données source.
ID secret	Choisissez l'ARN de votre secret dans Secrets Manager qui stocke les informations d'identification de la base de données cible.
Rôle IAM	Choisissez un rôle IAM pour fournir l'accès aux informations d'identification de la base de données cible dans Secrets Manager.
Règles de transformation	(Facultatif) Si vous créez un projet de migration pour la conversion de schéma DMS, choisissez Ajouter une règle de transformation pour configurer les règles de transformation. Les règles de transformation vous permettent de modifier les noms d'objets conformément à la règle que vous spécifiez. Pour plus d'informations, consultez Configuration des règles de transformation.

4. Choisissez Créer un projet de migration.

Après avoir AWS DMS créé votre projet de migration, vous pouvez utiliser ce projet dans DMS Schema Conversion ou dans des migrations de données homogènes. Pour commencer à utiliser votre projet de migration, sur la page Projets de migration, choisissez votre projet dans la liste.

Gestion des projets de migration dans AWS Database Migration Service

Une fois votre projet de migration créé, vous pouvez le modifier ou le supprimer. Par exemple, pour modifier le fournisseur de données source ou cible, modifiez votre projet de migration.

Vous ne pouvez modifier ou supprimer votre projet de migration qu'après avoir fermé les opérations de conversion de schéma ou de migration de données. Pour ce faire, choisissez votre projet de migration dans la liste, puis choisissez Conversion de schéma ou Migrations de données. Choisissez ensuite Fermer la conversion de schéma pour la conversion de schéma DMS et confirmez votre choix. Pour les migrations de données homogènes, choisissez votre migration de données, puis choisissez Arrêter dans le menu Actions. Après avoir modifié votre projet de migration, vous pouvez lancer la conversion de schéma ou recommencer la migration de données.

Pour modifier un projet de migration

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration, puis choisissez Modifier.
4. Mettez à jour le nom de votre projet, modifiez le profil d'instance ou modifiez les fournisseurs de données source et cible. Vous pouvez éventuellement ajouter ou modifier les règles de migration qui modifient les noms d'objets lors de la conversion.
5. Sélectionnez Enregistrer les modifications.

Pour supprimer un projet de migration

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Projets de migration. La page Projets de migration s'ouvre.
3. Choisissez votre projet de migration, puis choisissez Supprimer.
4. Choisissez Supprimer pour confirmer votre choix.

Bonnes pratiques pour AWS Database Migration Service

Pour utiliser AWS Database Migration Service (AWS DMS) plus efficacement, consultez les recommandations de cette section sur la méthode de migration de données la plus efficace.

Rubriques

- Planification de la migration pour AWS Database Migration Service
- Conversion de schémas
- Examen de la documentation publique d'AWS DMS
- Exécution d'une preuve de concept
- Améliorer les performances d'une migration AWS DMS
- Utilisation de votre propre serveur de noms sur site
- Migration des objets binaires volumineux (Large Binary Object, LOB)
- Amélioration des performances en cas de migration de tables volumineuses à l'aide du filtrage des lignes
- la réplication continue.
- Réduction de la charge sur votre base de données source
- Réduction des goulots d'étranglement sur la base de données cible
- Utilisation de la validation des données lors de la migration
- Surveillance de vos tâches AWS DMS à l'aide de métriques
- Événements et notifications
- Utilisation du journal des tâches pour résoudre des problèmes de migration
- Résolution des problèmes liés aux tâches de réplication à l'aide du voyage dans le temps
- Modification de l'utilisateur et du schéma pour une cible Oracle
- Modification des espaces de table d'index et de table pour une cible Oracle
- Mise à niveau d'une version d'instance de réplication
- Comprendre vos coûts de migration

Planification de la migration pour AWS Database Migration Service

Lors de la planification d'une migration de base de données à l'aide de AWS Database Migration Service, tenez compte des éléments suivants :

- Pour connecter vos bases de données source et cible à une instance de réplication AWS DMS, vous devez configurer un réseau. Pour ce faire, il suffit de connecter deux ressources AWS dans le même cloud privé virtuel (VPC) que votre instance de réplication. Il peut s'agir de configurations plus complexes telles que la connexion d'une base de données sur site à une instance de base de données Amazon RDS via un réseau privé virtuel (VPN). Pour plus d'informations, consultez Configurations réseau pour la migration de base de données.
- Points de terminaison sources et cibles : vous devez connaître les informations et les tables de la base de données source qui doivent être migrées vers la base de données cible. AWS DMS prend en charge la migration de schéma de base, y compris la création des tables et des clés primaires. Toutefois, AWS DMS ne crée pas automatiquement d'index secondaires, de clés étrangères, de comptes d'utilisateur, etc., dans la base de données cible. Selon vos moteurs de base de données source et cible, vous devrez peut-être configurer une journalisation supplémentaire ou modifier d'autres paramètres pour une base de données source ou cible. Pour plus d'informations, consultez Sources pour la migration des données et Cibles pour la migration des données.
- Migration de schéma et de code : AWS DMS n'effectue pas de conversion de schéma ni de code. Vous pouvez utiliser des outils tels qu'Oracle SQL Developer, MySQL Workbench ou pgAdmin III pour convertir votre schéma. Pour convertir un schéma existant en un moteur de base de données différent, vous pouvez utiliser l'AWS Schema Conversion Tool (AWS SCT). Il peut créer un schéma cible et générer et créer un schéma entier : tables, index, vues, etc. Vous pouvez également utiliser cet outil pour convertir des formats PL/SQL ou TSQL en PgSQL et autres. Pour plus d'informations sur AWS SCT, consultez le Guide de l'utilisateur AWS SCT.
- Types de données non pris en charge : veillez à pouvoir convertir les types de données sources en types de données équivalents pour la base de données cible. Pour plus d'informations sur les types de données pris en charge, consultez la section relative à la source ou à la cible pour votre magasin de données.
- Résultats des scripts d'assistance au diagnostic : lorsque vous planifiez votre migration, nous vous recommandons d'exécuter des scripts d'assistance au diagnostic. Les résultats de ces scripts vous permettront de trouver des informations avancées sur les échecs potentiels de migration.

  Si un script d'assistance est disponible pour la base de données, téléchargez-le à l'aide du lien figurant dans la rubrique du script correspondant dans la section suivante. Après avoir vérifié et examiné le script, vous pouvez l'exécuter conformément à la procédure décrite dans la rubrique relative au script dans votre environnement local. Lorsque l'exécution du script est terminée, vous pouvez passer en revue les résultats. Nous vous recommandons d'exécuter ces scripts comme première étape de tout effort de résolution des problèmes. Les résultats peuvent être utiles lorsque

vous travaillez avec une équipe AWS Support. Pour plus d'informations, consultez Utilisation de scripts d'assistance au diagnostic dans AWS DMS.

- Évaluations de prémigration : une évaluation de prémigration évalue les composants spécifiques d'une tâche de migration de base de données et vous aidera à identifier les problèmes susceptibles d'empêcher une tâche de migration de s'exécuter comme prévu. Cette évaluation vous permet d'identifier les problèmes potentiels avant d'exécuter une tâche nouvelle ou modifiée. Pour plus d'informations sur l'utilisation des évaluations de prémigration, consultez Activation et utilisation des évaluations de prémigration pour une tâche.

Conversion de schémas

AWS DMS n'effectue pas de conversion de schéma ou de code. Si vous souhaitez convertir un schéma existant en un moteur de base de données différent, vous pouvez utiliserAWS SCT. AWS SCT convertit vos objets, tables, index, vues et déclencheurs sources, ainsi que d'autres objets système au format DDL (Data Definition Language) cible. Vous pouvez également utiliser AWS SCT pour convertir la majeure partie de votre code d'application, comme PL/SQL ou TSQL, vers le langage cible équivalent.

Vous pouvez obtenir AWS SCT en le téléchargeant gratuitement à partir d'AWS. Pour plus d'informations sur AWS SCT, consultez le Guide de l'utilisateur AWS SCT.

Si vos points de terminaison source et cible se trouvent sur le même moteur de base de données, vous pouvez utiliser des outils tels qu'Oracle SQL Developer, MySQL Workbench ou PgAdmin 4 pour déplacer votre schéma.

Examen de la documentation publique d'AWS DMS

Nous vous recommandons vivement de consulter les pages de la documentation publique d'AWS DMS pour vos points de terminaison sources et cibles avant votre première migration. Cette documentation peut vous aider à identifier les conditions préalables à la migration et à comprendre les limitations actuelles avant de commencer. Pour plus d'informations, consultez Utilisation des points de terminaison AWS DMS.

Pendant la migration, la documentation publique peut vous aider à résoudre les éventuels problèmes liés à AWS DMS. Les pages de la documentation portant sur la résolution des problèmes peuvent vous aider à résoudre les problèmes courants en utilisant à la fois AWS DMS et les bases de

données de point de terminaison sélectionnées. Pour plus d'informations, consultez Résolution des problèmes liés aux tâches de migration AWS Database Migration Service.

Exécution d'une preuve de concept

Pour mieux détecter les problèmes liés à votre environnement dans les premières phases de la migration de la base de données, nous vous recommandons d'effectuer un petit test de migration. Cela peut également vous aider à définir une chronologie de migration plus réaliste. En outre, vous devrez peut-être effectuer un test de migration complet pour mesurer si AWS DMS peut gérer le débit de la base de données sur votre réseau. Pendant ce temps, nous vous recommandons de comparer et d'optimiser votre chargement complet initial et la réplication continue. Cela peut vous aider à comprendre la latence de votre réseau et à évaluer les performances globales.

À ce stade, vous avez également la possibilité de comprendre votre profil de données et la taille de la base de données, notamment les éléments suivants :

- Combien de tables ont une taille élevée, moyenne et réduite.
- Comment AWS DMS gère les conversions de types de données et de jeux de caractères.
- Combien de tables comportent des colonnes d'objets volumineux (LOB).
- Le temps nécessaire pour effectuer un test de migration.

Améliorer les performances d'une migration AWS DMS

Un certain nombre de facteurs affectent les performances de votre migration AWS DMS :

- Disponibilité des ressources sur la source.
- Débit du réseau disponible.
- Capacité des ressources du serveur de réplication.
- Capacité de la cible à ingérer les modifications.
- Type et distribution des données sources.
- Nombre d'objets à migrer.

Vous pouvez améliorer les performances en utilisant tout ou partie des bonnes pratiques mentionnées ci-après. La possibilité ou non d'utiliser ces pratiques dépend de votre cas d'utilisation spécifique. Vous pouvez trouver certaines limitations ci-dessous :

Provisionnement d'un serveur de réplication approprié

AWS DMS est un service géré qui s'exécute sur une instance Amazon EC2. Ce service se connecte à la base de données source, lit les données sources, formate les données en vue de leur consommation par la base de données cible et charge les données dans la base de données cible.

La majeure partie de ce traitement se passe dans la mémoire. Cependant, les transactions importantes peuvent nécessiter une mise en mémoire tampon sur disque. Les transactions mises en cache et les fichiers journaux sont également écrits sur le disque. Vous découvrirez dans les sections suivantes les éléments à prendre en compte lorsque vous choisissez votre serveur de réplication.

CPU

AWS DMS a été conçu pour des migrations hétérogènes, mais il prend également en charge les migrations homogènes. Pour effectuer une migration homogène, convertissez d'abord chaque type de données source vers le type de données AWS DMS équivalent. Convertissez ensuite chaque type de données AWS DMS vers le type de données cible. Vous trouverez des références relatives à ces conversions pour chaque moteur de base de données dans le Guide de l'utilisateur AWS DMS.

Pour qu'AWS DMS effectue ces conversions de manière optimale, le CPU doit être disponible au moment des conversions. La surcharge du CPU et le manque de ressources de CPU peuvent ralentir les migrations, ce qui peut entraîner d'autres effets secondaires.

Classe d'instances de réplication

Certaines des plus petites classes d'instance sont suffisantes pour tester le service ou effectuer de petites migrations. Si votre migration implique un grand nombre de tables, ou si vous prévoyez d'exécuter plusieurs tâches de réplication simultanées, envisagez d'utiliser une des instances les plus grandes. Une instance plus grande peut être une bonne idée car le service consomme une quantité importante de mémoire et de CPU.

Les instances de type T2 ont été conçues pour offrir des performances de référence modérées et la possibilité d'émettre en rafale pour atteindre des performances nettement supérieures si votre charge de travail l'exige. Elles sont prévues pour des charges de travail qui n'utilisent pas souvent ou de manière continue l'intégralité de la capacité de CPU, mais qui ont parfois besoin d'émettre en rafale. Les instances T2 conviennent parfaitement aux charges de travail à usage général, telles que les serveurs web, les environnements de développement et les petites bases de données. Si vous résolvez les problèmes liés à une migration lente et que vous utilisez un

type d'instance T2, vérifiez la métrique hôte d'utilisation de CPU. Elle peut vous indiquer si vous dépassez la référence de ce type d'instance.

Les classes d'instances C4 ont été conçues de façon à proposer le plus haut niveau de performances de processeur pour les charges de travail informatiques intensives. Elles offrent des performances nettement supérieures en termes de nombre de paquets par seconde (PPS), et réduisent l'instabilité et la latence réseau. AWS DMS peut nécessiter une utilisation intensive de CPU, en particulier lors de migrations et de réplications hétérogènes, telles que la migration d'Oracle vers PostgreSQL. Les instances C4 peuvent être un choix judicieux pour ces situations.

Les classes d'instances R4 sont optimisées pour la mémoire pour les charges de travail consommatrices de mémoire. Les migrations et réplications continues des systèmes de transactions à haut débit utilisant AWS DMS peuvent parfois consommer de grandes quantités de ressources de CPU et de mémoire. Les instances R4 incluent davantage de mémoire par vCPU.

Prise en charge d'AWS DMS pour les classes d'instances R5 et C5

Les classes d'instances R5 sont des instances à mémoire optimisée conçues pour fournir des performances rapides pour les charges de travail qui traitent des jeux de données volumineux en mémoire. Les migrations et réplications continues des systèmes de transactions à haut débit utilisant AWS DMS peuvent parfois consommer de grandes quantités de ressources de CPU et de mémoire. Les instances R5 fournissent 5 % de mémoire supplémentaire par vCPU par rapport aux instances R4, et les instances de plus grande taille fournissent 768 Gio de mémoire. En outre, les instances R5 offrent une amélioration de 10 % du prix par Gio et une augmentation d'environ 20 % des performances de CPU par rapport aux instances R4.

Les classes d'instances C5 sont optimisées pour les charges de travail intensives en termes de calcul et offrent des performances élevées et rentables à un faible ratio prix/calcul. Elles améliorent considérablement les performances du réseau. L'adaptateur réseau élastique (ENA) fournit aux instances C5 jusqu'à 25 Gbit/s de bande passante réseau et jusqu'à 14 Gbit/s de bande passante dédiée à Amazon EBS. AWS DMS peut nécessiter une utilisation intensive de CPU, en particulier lors de migrations et de réplications hétérogènes, telles que la migration d'Oracle vers PostgreSQL. Les instances C5 peuvent être un choix judicieux pour de telles situations.

Stockage

En fonction de la classe d'instances, votre serveur de réplication est fourni avec un stockage de données de 50 Go ou 100 Go. Ce stockage est utilisé pour les fichiers journaux et toutes les modifications mises en cache collectées pendant le chargement. Si votre système source

est occupé ou accepte des transactions importantes, vous devrez peut-être augmenter votre capacité de stockage. Si vous exécutez plusieurs tâches sur le serveur de réplication, vous pouvez également avoir besoin d'une augmentation de la capacité de stockage. Toutefois, la quantité par défaut est généralement suffisante.

Tous les volumes de stockage dans AWS DMS sont des disques durs à état solide (SSD) GP2 ou à usage général. Les volumes GP2 sont dotés d'une performance de base de trois opérations d'E/S par seconde (IOPS), avec la capacité d'émettre en rafale jusqu'à 3 000 IOPS sur la base de crédits. En règle générale, vérifiez les métriques `ReadIOPS` et `WriteIOPS` pour l'instance de réplication. Assurez-vous que la somme de ces valeurs ne dépasse pas les performances de base pour ce volume.

Multi-AZ

Le choix d'une instance multi-AZ peut protéger votre migration contre les défaillances de stockage. La plupart des migrations sont transitoires et ne sont pas destinées à s'exécuter pendant de longues périodes. Si vous utilisez AWS DMS à des fins de réplication continue, le choix d'une instance multi-AZ peut améliorer votre disponibilité en cas de problème de stockage.

Lorsque vous utilisez une seule instance de réplication AZ ou multi-AZ pendant un CHARGEMENT COMPLET et qu'un basculement ou un remplacement d'hôte se produit, la tâche de chargement complet est censée échouer. Vous pouvez redémarrer la tâche à partir du point de défaillance pour les tables restantes qui ne sont pas terminées ou qui présentent un état d'erreur.

Chargement de plusieurs tables en parallèle

Par défaut, AWS DMS charge huit tables à la fois. Vous pouvez voir une amélioration des performances en augmentant ce nombre légèrement lorsque vous utilisez un serveur de réplication très volumineux, par exemple un dms.c4.xlarge ou une instance plus grande. Cependant, à un moment donné, l'augmentation de ce parallélisme réduit les performances. Si votre serveur de réplication est relativement petit, comme un dms.t2.medium, nous vous recommandons de réduire le nombre de tables chargées en parallèle.

Pour modifier ce nombre dans la AWS Management Console, ouvrez la console, choisissez Tâches, choisissez de créer ou modifier une tâche, puis choisissez Paramètres avancés. Sous Tuning Settings (Paramètres de réglage), modifiez l'option Maximum number of tables to load in parallel (Nombre maximal de tables à charger en parallèle).

Pour modifier ce nombre à l'aide d'AWS CLI, modifiez le paramètre `MaxFullLoadSubTasks` sous `TaskSettings`.

Utilisation du chargement complet en parallèle

Vous pouvez utiliser un chargement parallèle à partir de sources Oracle, Microsoft SQL Server, MySQL, Sybase et IBM Db2 LUW sur la base de partitions et de sous-partitions. Cela peut réduire la durée globale du chargement complet. En outre, lors de l'exécution d'une tâche de migration AWS DMS, vous pouvez accélérer la migration de tables volumineuses ou partitionnées. Pour ce faire, divisez la table en segments et chargez les segments en parallèle dans la même tâche de migration.

Pour utiliser un chargement parallèle, créez une règle de mappage de tables de type `table-settings` avec l'option `parallel-load`. Dans la règle `table-settings`, spécifiez les critères de sélection de la ou des tables que vous souhaitez charger en parallèle. Pour spécifier les critères de sélection, définissez l'élément `type` pour `parallel-load` sur l'une des valeurs suivantes :

- `partitions-auto`
- `subpartitions-auto`
- `partitions-list`
- `ranges`
- `none`

Pour plus d'informations sur ces paramètres, consultez Règles des paramètres de table et de collection et opérations.

Utilisation d'index, de déclencheurs et de contraintes d'intégrité référentielle

Les index, les déclencheurs et les contraintes d'intégrité référentielle peuvent affecter vos performances de migration et provoquer l'échec de votre migration. La manière dont ces éléments affectent la migration dépend du fait que votre tâche de réplication soit une tâche de chargement complet ou une tâche de réplication continue (capture des données de modification ou CDC).

Pour une tâche de chargement complet, nous vous recommandons de supprimer les index de clé primaire, les index secondaires, les contraintes d'intégrité référentielle et les déclencheurs DML (Data Manipulation Language). Vous pouvez aussi retarder leur création jusqu'à ce que les tâches de chargement complet soient terminées. Vous n'avez pas besoin des index au cours d'une tâche de chargement complet et les index occasionnent des frais de maintenance s'ils sont présents. Étant donné que la tâche de chargement complet charge des groupes de tables en une seule fois, les contraintes d'intégrité référentielle sont violées. De même, l'insertion, la mise à jour et la suppression de déclencheurs peuvent entraîner des erreurs, par exemple, si une insertion de ligne est déclenchée pour une table préalablement chargée en bloc. D'autres types

de déclencheurs affectent également les performances en raison du traitement supplémentaire qu'ils entraînent.

Si vos volumes de données sont relativement petits et si le temps de migration supplémentaire ne vous pose pas de problème, vous pouvez créer des index de clé primaire et des index secondaires avant une tâche de chargement complet. Désactivez toujours les contraintes d'intégrité référentielle et les déclencheurs.

Pour une tâche de chargement complet + CDC, nous vous recommandons d'ajouter les index secondaires avant la phase CDC. Comme AWS DMS utilise la réplication logique, veillez à ce que les index secondaires qui prennent en charge les opérations DML soient en place pour empêcher les analyses de tables complètes. Vous pouvez suspendre la tâche de réplication avant la phase CDC afin de construire des index et de créer des contraintes d'intégrité référentielle avant de redémarrer la tâche.

Vous devez activer les déclencheurs juste avant le basculement.

Désactivation des sauvegardes et de la journalisation des transactions

Lors de la migration vers une base de données Amazon RDS, il est judicieux de désactiver les sauvegardes et Multi-AZ sur la cible jusqu'à ce que vous soyez prêt à effectuer le basculement. De même, lors de la migration vers des systèmes autres qu'Amazon RDS, il est généralement judicieux de désactiver la journalisation sur la cible jusqu'à ce que le basculement soit terminé.

Utilisation de plusieurs tâches

L'utilisation de plusieurs tâches pour une seule migration peut permettre d'améliorer les performances. Si vous avez des ensembles de tables qui n'interviennent pas dans des transactions communes, vous pouvez peut-être diviser votre migration en plusieurs tâches. La cohérence transactionnelle est maintenue au sein d'une tâche. Il est donc important que les tables de tâches distinctes n'interviennent pas dans des transactions communes. De plus, chaque tâche lisant indépendamment le flux de transactions, veillez à ne pas trop utiliser la base de données source.

Vous pouvez utiliser plusieurs tâches pour créer des flux de réplication distincts. Vous pouvez ainsi paralléliser les lectures sur la source, les processus sur l'instance de réplication et les écritures dans la base de données cible.

Optimisation du traitement des modifications

Par défaut, AWS DMS traite les modifications en mode transactionnel, ce qui préserve l'intégrité transactionnelle. Si vous pouvez vous permettre des écarts temporaires dans l'intégrité

transactionnelle, vous pouvez utiliser l'option d'application par lots optimisée à la place. Cette option regroupe efficacement les transactions et les applique par lots pour plus d'efficacité. L'utilisation de l'option d'application optimisée par lots enfreint presque toujours les contraintes d'intégrité référentielle. Nous vous recommandons donc de désactiver ces contraintes pendant le processus de migration et de les réactiver dans le cadre du processus de basculement.

Utilisation de votre propre serveur de noms sur site

Généralement, une instance de réplication AWS DMS utilise le résolveur DNS (Domain Name System) dans une instance Amazon EC2 pour résoudre les points de terminaison de domaine. Toutefois, vous pouvez utiliser votre propre serveur de noms sur site pour résoudre certains points de terminaison si vous utilisez Amazon Route 53 Resolver. Cet outil vous permet d'effectuer des requêtes entre des sites locaux et AWS en utilisant des points de terminaison entrants et sortants, des règles de transfert et une connexion privée. Les avantages de l'utilisation d'un serveur de noms sur site incluent une sécurité améliorée et une facilité d'utilisation derrière un pare-feu.

Si vous avez des points de terminaison entrants, vous pouvez utiliser des requêtes DNS provenant de sites locaux pour résoudre les domaines hébergés par AWS. Pour configurer les points de terminaison, attribuez des adresses IP dans chaque sous-réseau auquel vous souhaitez fournir un résolveur. Pour établir une connectivité entre votre infrastructure DNS sur site et AWS, utilisez AWS Direct Connect ou un réseau privé virtuel (VPN).

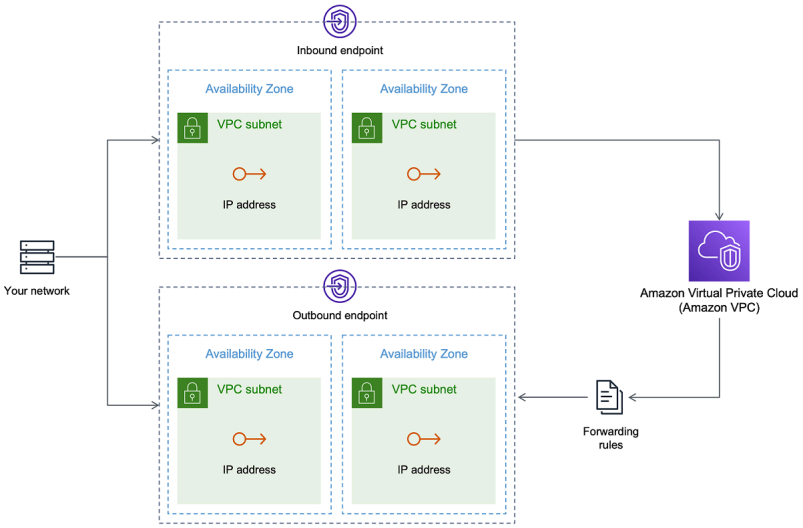
Les points de terminaison sortants se connectent à votre serveur de noms sur site. Le serveur de noms n'accorde l'accès qu'aux adresses IP incluses dans une liste d'autorisation et définies pour un point de terminaison sortant. L'adresse IP de votre serveur de noms est l'adresse IP cible. Lorsque vous choisissez un groupe de sécurité pour un point de terminaison sortant, choisissez le même groupe de sécurité que celui utilisé par l'instance de réplication.

Pour transférer des domaines sélectionnés vers le serveur de noms, utilisez des règles de transfert. Un point de terminaison sortant peut gérer plusieurs règles de transfert. La portée de la règle de transfert est votre cloud privé virtuel (VPC). En utilisant une règle de transfert associée à un VPC, vous pouvez provisionner une section logiquement isolée du Cloud AWS. À partir de cette section logiquement isolée, vous pouvez lancer des ressources AWS dans un réseau virtuel.

Vous pouvez configurer les domaines hébergés au sein de votre infrastructure DNS sur site en tant que règles de transfert conditionnel configurant des requêtes DNS sortantes. Lorsqu'une requête est effectuée sur l'un de ces domaines, les règles déclenchent une tentative de transfert des demandes

DNS vers les serveurs qui ont été configurés avec ces règles. Encore une fois, une connexion privée via AWS Direct Connect ou un VPN est nécessaire.

Le schéma suivant illustre l'architecture de Route 53 Resolver.



Pour plus d'informations sur le résolveur DNS de Route 53, consultez Mise en route avec Route 53 Resolver dans le Guide du développeur Amazon Route 53.

Utilisation d'Amazon Route 53 Resolver avec AWS DMS

Vous pouvez créer un serveur de noms sur site pour permettre à AWS DMS de résoudre les points de terminaison à l'aide d'Amazon Route 53 Resolver.

Pour créer un serveur de noms sur site pour AWS DMS basé sur Route 53

1. Connectez-vous à AWS Management Console et ouvrez la console Route 53 à l'adresse https://console.aws.amazon.com/route53/.
2. Dans la console Route 53, choisissez la région AWS dans laquelle vous souhaitez configurer votre résolveur Route 53. Le résolveur Route 53 est spécifique à une région.
3. Choisissez le sens de requête : entrant, sortant ou les deux.
4. Fournissez votre configuration de requête entrante :

    a. Entrez un nom de point de terminaison et choisissez un VPC.

    b. Affectez un ou plusieurs sous-réseaux à partir du VPC (par exemple, choisissez-en deux pour assurer la disponibilité).

    c. Attribuez des adresses IP spécifiques à utiliser comme points de terminaison ou laissez le résolveur Route 53 les attribuer automatiquement.

5. Créez une règle pour votre domaine local afin que les charges de travail à l'intérieur du VPC puissent router les requêtes DNS vers votre infrastructure DNS.
6. Entrez une ou plusieurs adresses IP pour vos serveurs DNS sur site.
7. Soumettez votre règle.

Quand tout est créé, votre VPC est associé à vos règles entrantes et sortantes et peut commencer à router le trafic.

Pour plus d'informations sur le résolveur Route 53, consultez Mise en route avec Route 53 Resolver dans le Guide du développeur Amazon Route 53.

Migration des objets binaires volumineux (Large Binary Object, LOB)

En général, AWS DMS migre les données LOB en deux phases :

1. AWS DMS crée une nouvelle ligne dans la table cible et remplit cette ligne avec toutes les données à l'exception de la valeur LOB associée.
2. AWS DMS met à jour la ligne dans la table cible avec les données LOB.

Ce processus de migration pour les objets LOB exige pendant la migration que toutes les colonnes LOB de la table cible autorisent les valeurs Null. Il en est ainsi même si les colonnes LOB n'autorisent pas les valeurs Null dans la table source. Si AWS DMS crée les tables cible, elle définit les colonnes LOB sur le type nullable par défaut. Dans certains cas, vous pouvez créer les tables cibles à l'aide d'un autre mécanisme, tel que l'importation ou l'exportation. Dans ce cas, assurez-vous que les colonnes LOB autorisent les valeurs Null avant de commencer la tâche de migration.

Cette exigence n'a qu'une exception. Supposons que vous effectuez une migration homogène d'une source Oracle vers une cible Oracle et que vous choisissez Limited Lob mode (Mode LOB limité). Dans ce cas, la ligne entière est remplie en une seule fois, y compris les valeurs LOB. Dans ce cas, AWS DMS peut créer les colonnes LOB de la table cible avec des contraintes n'autorisant pas la valeur Null, si nécessaire.

Utilisation du mode LOB limité

AWS DMS utilise deux méthodes qui assurent un équilibre entre performances et commodité lorsque votre migration contient des valeurs LOB :

1. Le paramètre Limited LOB mode (Mode LOB limité) permet de migrer toutes les valeurs LOB jusqu'à une taille limite spécifiée par l'utilisateur (la valeur par défaut est 32 Ko). Les valeurs LOB supérieures à la taille limite doivent être migrées manuellement. Le paramètre Limited LOB mode (Mode LOB limité) (paramètre par défaut pour toutes les tâches de migration) offre généralement les meilleures performances. Toutefois, vous devez vous assurer que le paramètre Taille de LOB maximale est correctement défini. Définissez ce paramètre sur la plus grande taille de LOB pour toutes vos tables.
2. Le paramètre Full LOB mode (Mode LOB complet) migre toutes les données LOB de vos tables, quelle que soit leur taille. Le paramètre Full LOB mode (Mode LOB complet) permet de déplacer toutes les données LOB de vos tables, mais le processus peut avoir un impact significatif sur les performances.

Pour certains moteurs de base de données, tels que PostgreSQL, AWS DMS traite les types de données JSON comme des données LOB. Si vous avez choisi Mode LOB limité, assurez-vous que l'option Taille de LOB maximale est définie sur une valeur qui n'entraîne pas la troncation des données JSON.

AWS DMS prend en charge intégralement l'utilisation des types de données LOB (BLOB, CLOB et NCLOB). Les points de terminaison sources suivants prennent en charge intégralement les LOB :

- Oracle
- Microsoft SQL Server
- ODBC

Les points de terminaison cibles suivants prennent en charge intégralement les LOB :

- Oracle
- Microsoft SQL Server

Le point de terminaison cible suivant offre une prise en charge limitée des LOB. Vous ne pouvez pas utiliser une taille LOB illimitée pour ce point de terminaison cible.

- Amazon Redshift
- Amazon S3

Pour les points de terminaison prennent en charge les LOB intégralement, vous pouvez également définir une limite de taille pour les types de données LOB.

Amélioration des performances des objets LOB

Lors de la migration de données LOB, vous pouvez spécifier les différents paramètres d'optimisation LOB suivants.

Paramètres LOB par table

Les paramètres LOB par table vous permettent de remplacer les paramètres LOB au niveau d'une tâche pour une partie ou la totalité de vos tables. Pour ce faire, définissez `lob-settings` dans votre règle `table-settings`. Voici un exemple de table qui inclut des valeurs LOB élevées.

```
SET SERVEROUTPUT ON
CREATE TABLE TEST_CLOB
(
ID NUMBER,
C1 CLOB,
C2 VARCHAR2(4000)
);
DECLARE
bigtextstring CLOB := '123';
iINT;
BEGIN
WHILE Length(bigtextstring) <= 60000 LOOP
bigtextstring := bigtextstring || '00000000000000000000000000000000';
END LOOP;
INSERT INTO TEST_CLOB (ID, C1, C2) VALUES (0, bigtextstring,'AnyValue');
END;
/
SELECT * FROM TEST_CLOB;
COMMIT
```

Créez ensuite une tâche de migration et modifiez le traitement des objets LOB pour votre table à l'aide de la nouvelle règle `lob-settings`. La valeur `bulk-max-siz` détermine la taille de LOB maximale (Ko). Il est tronqué s'il dépasse la taille spécifiée.

```
{
 "rules": [{
   "rule-type": "selection",
   "rule-id": "1",
```

```
   "rule-name": "1",
   "object-locator": {
    "schema-name": "HR",
    "table-name": "TEST_CLOB"
   },
   "rule-action": "include"
  },
  {
   "rule-type": "table-settings",
   "rule-id": "2",
   "rule-name": "2",
   "object-locator": {
    "schema-name": "HR",
    "table-name": "TEST_CLOB"
   },
   "lob-settings": {
    "mode": "limited",
    "bulk-max-size": "16"
   }
  }
 ]
}
```

Même si cette tâche AWS DMS est créée avec `FullLobMode : true`, les paramètres LOB par table indiquent à AWS DMS de tronquer les données LOB dans cette table particulière à 16 000. Vous pouvez examiner les journaux de tâches pour le confirmer.

```
721331968: 2018-09-11T19:48:46:979532 [SOURCE_UNLOAD] W: The value of column 'C' in
 table
'HR.TEST_CLOB' was truncated to length 16384
```

Paramètres LOB en ligne

Lorsque vous créez une tâche AWS DMS, le mode LOB détermine la manière de traiter les objets LOB.

Avec le mode LOB complet et le mode LOB limité, chacun a ses avantages et ses inconvénients propres. Le mode LOB en ligne combine les avantages du mode LOB complet et du mode LOB limité.

Vous pouvez utiliser le mode LOB en ligne lorsque vous devez répliquer des objets LOB de petite taille et d'autres de grande taille, et que la plupart des objets LOB sont de petite taille. Lorsque

vous choisissez cette option, pendant le chargement complet, la tâche AWS DMS transfère les petits objets LOB en ligne, ce qui est plus efficace. La tâche AWS DMS transfère les objets LOB volumineux en effectuant une recherche à partir de la table source.

Pendant le traitement des modifications, les objets LOB de petite et de grande taille sont répliqués en effectuant une recherche à partir de la table source.

Lorsque vous utilisez le mode LOB en ligne, la tâche AWS DMS vérifie toutes les tailles de LOB pour déterminer celles à transférer en ligne. Les objets LOB supérieurs à la taille spécifiée sont répliqués en mode LOB complet. Par conséquent, si vous savez que la plupart des objets LOB sont supérieurs au paramètre spécifié, il est préférable de ne pas utiliser cette option. Autorisez plutôt une taille de LOB illimitée.

Vous configurez cette option à l'aide d'un attribut dans les paramètres des tâches, `InlineLobMaxSize`, qui est disponible uniquement quand `FullLobMode` a pour valeur `true`. La valeur par défaut pour `InlineLobMaxSize` est 0 et la plage est de 1 à 102 400 kilo-octets (100 Mo).

Par exemple, vous pouvez utiliser les paramètres de tâche AWS DMS suivants. Dans ce cas, affecter à `InlineLobMaxSize` la valeur 5 entraîne le transfert en ligne de tous les objets LOB inférieurs ou égaux à 5 Kio (5 120 octets).

```
{
  "TargetMetadata": {
  "TargetSchema": "",
  "SupportLobs": true,
  "FullLobMode": true,
  "LobChunkSize": 64,
  "LimitedSizeLobMode": false,
  "LobMaxSize": 32,
  "InlineLobMaxSize": 5,
  "LoadMaxFileSize": 0,
  "ParallelLoadThreads": 0,
  "ParallelLoadBufferSize":0,
  "BatchApplyEnabled": false,
  "TaskRecoveryTableEnabled": false},
  . . .
}
```

Amélioration des performances en cas de migration de tables volumineuses à l'aide du filtrage des lignes

Pour améliorer les performances lors de la migration d'une table volumineuse, divisez la migration en plusieurs tâches. Pour diviser la migration en plusieurs tâches à l'aide du filtrage des lignes, utilisez une clé ou une clé de partition. Par exemple, si vous avez un ID de clé primaire entier compris entre 1 et 8 000 000, vous pouvez créer huit tâches à l'aide du filtrage de lignes et migrer 1 million d'enregistrements dans chacune des tâches.

Pour appliquer le filtrage des lignes dans la console :

1. Ouvrez la AWS Management Console.
2. Choisissez Tâches et créez une nouvelle tâche.
3. Choisissez l'onglet Mappages de table, puis développez Règles de sélection.
4. Choisissez Ajouter une nouvelle règle de sélection. Vous pouvez désormais ajouter un filtre de colonnes avec une condition d'infériorité ou d'égalité, de supériorité ou d'égalité, d'égalité, ou de plage entre deux valeurs. Pour plus d'informations sur le filtrage des colonnes, consultez Spécification des règles de sélection de table et de transformation à partir de la console.

Si vous avez une table partitionnée de grande taille qui est partitionnée par date, vous pouvez migrer les données en fonction de la date. Par exemple, supposons que vous avez une table partitionnée par mois, et que seules les données du mois en cours sont mises à jour. Dans ce cas, vous pouvez créer une tâche de chargement complet pour chaque partition mensuelle statique et créer une tâche de chargement complet + CDC pour la partition actuellement mise à jour.

Si votre table possède une clé primaire à colonne unique ou un index unique, vous pouvez demander à votre tâche AWS DMS de segmenter la table en utilisant un chargement parallèle de type plage pour charger les données en parallèle. Pour plus d'informations, consultez Règles des paramètres de table et de collection et opérations.

la réplication continue.

AWS DMS permet la réplication continue des données, en maintenant la synchronisation entre les bases de données source et cible. Il réplique uniquement une quantité limitée d'instructions de langage de définition de données (DDL). AWS DMS ne propage pas les éléments tels que les

index, les utilisateurs, les privilèges, les procédures stockées et les autres modifications de base de données qui ne sont pas directement liées aux données de table.

Si vous envisagez d'utiliser la réplication continue, définissez l'option Multi-AZ quand vous créez votre instance de réplication. En choisissant Multi-AZ, vous obtenez la prise en charge de la haute disponibilité et du basculement pour l'instance de réplication. Toutefois, cette option peut avoir un impact sur les performances et ralentir la réplication lors de l'application des modifications au système cible.

Avant de mettre à niveau vos bases de données source et cible, nous vous recommandons d'arrêter toutes les tâches AWS DMS en cours d'exécution sur ces bases de données. Reprenez ces tâches une fois les mises à niveau terminées.

Au cours de la réplication continue, il est essentiel d'identifier la bande passante du réseau entre votre système de base de données source et votre instance de réplication AWS DMS. Assurez-vous que le réseau n'est pas à l'origine de goulots d'étranglement lors de la réplication continue.

Il est également important d'identifier le taux de génération de journaux de modifications et d'archivage par heure sur votre système de base de données source. Cela peut vous aider à comprendre le débit que vous pouvez obtenir lors d'une réplication continue.

Réduction de la charge sur votre base de données source

AWS DMS utilise certaines des ressources de votre base de données source. Lors d'une tâche de chargement complet, AWS DMS effectue une analyse complète de la table source pour chaque table traitée en parallèle. De plus, chaque tâche que vous créez dans le cadre d'une migration interroge la source pour connaître les modifications apportées dans le cadre du processus CDC. Pour qu'AWS DMS puisse effectuer le processus CDC pour certaines sources, telles qu'Oracle, vous pouvez être amené à augmenter la quantité de données écrites dans le journal des modifications de votre base de données.

Si vous constatez que vous surchargez la base de données source, réduisez le nombre de tâches ou de tables pour chaque tâche de la migration. Chaque tâche récupère les modifications source de manière indépendante et, par conséquent, le regroupement des tâches peut diminuer la charge de travail de capture des modifications.

Réduction des goulots d'étranglement sur la base de données cible

Pendant la migration, essayez de supprimer tous les processus qui se disputent les ressources d'écriture sur la base de données cible :

- Désactivez les déclencheurs inutiles.
- Désactivez les index secondaires lors du chargement initial et réactivez-les ultérieurement pendant la réplication continue.
- Avec les bases de données Amazon RDS, il est conseillé de désactiver les sauvegardes et le mode multi-AZ jusqu'au basculement.
- Lors de la migration vers des systèmes non-RDS, il est judicieux de désactiver toute journalisation sur la cible jusqu'au basculement.

Utilisation de la validation des données lors de la migration

Pour garantir que vos données ont été migrées avec précision de la source vers la cible, nous vous recommandons vivement d'utiliser la validation des données. Si vous activez la validation des données pour une tâche, AWS DMS commence à comparer les données de la source et celles de la cible immédiatement après la réalisation d'un chargement complet pour une table.

La validation de données fonctionne avec les bases de données suivantes lorsque AWS DMS les prend en charge en tant que points de terminaison source et cible :

- Oracle
- PostgreSQL
- MySQL
- MariaDB
- Microsoft SQL Server
- Amazon Aurora MySQL-Compatible Edition
- Amazon Aurora PostgreSQL-Compatible Edition
- IBM Db2 LUW
- Amazon Redshift

Pour plus d'informations, consultez Validation des données AWS DMS.

Surveillance de vos tâches AWS DMS à l'aide de métriques

Plusieurs options s'offrent à vous pour surveiller les métriques relatives à vos tâches à l'aide de la console AWS DMS :

Métriques d'hôte

Vous pouvez trouver les métriques d'hôte dans l'onglet CloudWatch Metrics pour chaque instance de réplication spécifique. Vous pouvez y surveillez si le dimensionnement de votre instance de réplication est approprié.

Métriques de tâches de réplication

Les métriques relatives aux tâches de réplication, y compris les modifications entrantes et validées, ainsi que le temps de latence entre l'hôte de réplication et les bases de données source/cible sont disponibles dans l'onglet CloudWatch des métriques pour chaque tâche en particulier.

Métriques de table

Vous pouvez trouver des métriques de table individuelles dans l'onglet Statistiques de table pour chaque tâche individuelle. Ces métriques incluent les nombres suivants :

- Lignes chargées au cours du chargement complet.
- Insertions, mises à jour et suppressions depuis le début de la tâche.
- Opérations DDL depuis le début de la tâche.

Pour plus d'informations sur la surveillance des métriques, consultez Surveillance des tâches AWS DMS.

Événements et notifications

AWS DMS utilise Amazon SNS pour fournir des notifications lorsqu'un événement AWS DMS se produit, par exemple la création ou la suppression d'une instance de réplication. Vous pouvez utiliser ces notifications sous n'importe quelle forme prise en charge par Amazon SNS pour une région AWS. Il peut s'agir d'e-mails, de messages texte ou d'appels vers un point de terminaison HTTP.

Pour plus d'informations, consultez Utilisation des événements et des notifications Amazon SNS dans AWS Database Migration Service.

Utilisation du journal des tâches pour résoudre des problèmes de migration

Dans certains cas, AWS DMS peut rencontrer des problèmes pour lesquels des avertissements ou des messages d'erreur s'affichent uniquement dans le journal des tâches. En particulier, les problèmes de troncature des données ou de rejet de lignes en raison de violations de clé étrangère sont écrits uniquement dans le journal des tâches. Par conséquent, veillez à examiner le journal des tâches lorsque vous migrez une base de données. Pour consulter le journal des tâches, configurez Amazon dans CloudWatch le cadre de la création des tâches.

Pour plus d'informations, consultez la section Surveillance des tâches de réplication à l'aide d'Amazon CloudWatch.

Résolution des problèmes liés aux tâches de réplication à l'aide du voyage dans le temps

Pour résoudre les problèmes de migration AWS DMS, vous pouvez utiliser le voyage dans le temps. Pour plus d'informations sur le voyage dans le temps, consultez Paramètres de tâche de voyage dans le temps.

Lorsque vous utilisez le voyage dans le temps, tenez compte des considérations suivantes :

- Pour éviter de surcharger une instance de réplication DMS, activez le voyage dans le temps uniquement pour les tâches nécessitant un débogage.
- Lorsque vous utilisez le voyage dans le temps pour résoudre des problèmes liés à des tâches de réplication susceptibles de s'exécuter pendant plusieurs jours, surveillez les métriques des instances de réplication pour détecter les surcharges de ressources. Cette approche s'applique particulièrement dans les cas où des charges de transactions élevées s'exécutent sur les bases de données sources pendant de longues périodes. Pour en savoir plus, consultez Surveillance des tâches AWS DMS.
- Lorsque le paramètre de tâche de voyage dans le temps `EnableRawData` a pour valeur `true`, l'utilisation de la mémoire des tâches pendant la réplication DMS peut être plus élevée que lorsque le voyage dans le temps n'est pas activé. Si vous activez le voyage dans le temps pendant de longues périodes, surveillez votre tâche.
- À l'heure actuelle, vous pouvez activer le voyage dans le temps seulement au niveau d'une tâche. Les modifications apportées à toutes les tables sont enregistrées dans les journaux de voyage

dans le temps. Pour résoudre les problèmes liés à des tables spécifiques dans une base de données présentant un volume de transactions élevé, créez une tâche distincte.

Modification de l'utilisateur et du schéma pour une cible Oracle

Quand vous utilisez Oracle en tant que cible, AWS DMS migre les données vers le schéma détenu par l'utilisateur du point de terminaison cible.

Par exemple, supposons que vous migrez un schéma nommé `PERFDATA` vers un point de terminaison cible Oracle, et que le nom de l'utilisateur du point de terminaison cible est `MASTER`. AWS DMS se connecte à la cible Oracle en tant que `MASTER` et remplit le schéma `MASTER` avec les objets de base de données provenant de `PERFDATA`.

Pour remplacer ce comportement, fournissez une transformation de schéma. Par exemple, pour migrer les objets du schéma `PERFDATA` vers un schéma `PERFDATA` au niveau du point de terminaison cible, utilisez la transformation suivante.

```
{
    "rule-type": "transformation",
    "rule-id": "2",
    "rule-name": "2",
    "object-locator": {
        "schema-name": "PERFDATA"
    },
    "rule-target": "schema",
    "rule-action": "rename",
    "value": "PERFDATA"
}
```

Pour plus d'informations sur les transformations, consultez Spécification des règles de sélection de table et de transformation à l'aide de JSON.

Modification des espaces de table d'index et de table pour une cible Oracle

Lorsque vous utilisez Oracle en tant que cible, AWS DMS migre toutes les tables et tous les index vers l'espace de table par défaut de la cible. Par exemple, supposons que votre source est un moteur de base de données autre qu'Oracle. Toutes les tables et tous les index cibles sont migrés vers le même espace de table par défaut.

Pour remplacer ce comportement, fournissez les transformations d'espace de table correspondantes. Par exemple, supposons que vous souhaitez migrer des tables et des index vers des espaces de table de table et d'index dans la cible Oracle qui sont nommés après le schéma dans la source. Dans ce cas, vous pouvez utiliser des transformations similaires à ce qui suit : Ici, le schéma dans la source est nommé `INVENTORY` et les espaces de table d'index et de table correspondants dans la cible sont nommés `INVENTORYTBL` et `INVENTORYIDX`.

```
{
   "rule-type": "transformation",
   "rule-id": "3",
   "rule-name": "3",
   "rule-action": "rename",
   "rule-target": "table-tablespace",
   "object-locator": {
      "schema-name": "INVENTORY",
      "table-name": "%",
      "table-tablespace-name": "%"
   },
   "value": "INVENTORYTBL"
},
{
   "rule-type": "transformation",
   "rule-id": "4
   "rule-name": "4",
   "rule-action": "rename",
   "rule-target": "index-tablespace",
   "object-locator": {
      "schema-name": "INVENTORY",
      "table-name": "%",
      "index-tablespace-name": "%"
   },
   "value": "INVENTORYIDX"
}
```

Pour plus d'informations sur les transformations, consultez Spécification des règles de sélection de table et de transformation à l'aide de JSON.

Si Oracle est à la fois source et cible, vous pouvez conserver les affectations d'espaces de table d'index et de table existantes en définissant l'attribut de connexion supplémentaire de la source Oracle, `enableHomogenousTablespace=true`. Pour plus d'informations, consultez Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS.

Mise à niveau d'une version d'instance de réplication

AWS publie régulièrement de nouvelles versions du logiciel de moteur de réplication AWS DMS avec de nouvelles fonctionnalités et améliorations des performances. Chaque version du logiciel de moteur de réplication a son propre numéro de version. Il est essentiel de tester la version existante de votre instance de réplication AWS DMS exécutant une charge de travail de production avant de mettre à niveau votre instance de réplication vers une version ultérieure. Pour plus d'informations sur les mises à niveau des versions disponibles, consultez AWS Notes de mise à jour du DMS.

Comprendre vos coûts de migration

AWS Database Migration Service vous permet de migrer des bases de données vers AWS facilement, en toute sécurité et à moindre coût. Vous ne payez que pour vos instances de réplication et pour tout stockage de journaux supplémentaire. Chaque instance de migration de base de données inclut un espace de stockage suffisant pour l'espace d'échange, les journaux de réplication et le cache de données pour la plupart des réplications. De plus, le transfert de données entrantes est gratuit.

Il se peut que vous ayez besoin de plus de ressources lors du chargement initial ou lors d'un pic de chargement. Vous pouvez surveiller de près l'utilisation des ressources des instances de réplication à l'aide des métriques CloudWatch. Vous pouvez ensuite augmenter ou réduire la taille des instances de réplication en fonction de l'utilisation.

Pour plus d'informations sur l'estimation de vos coûts de migration, consultez :

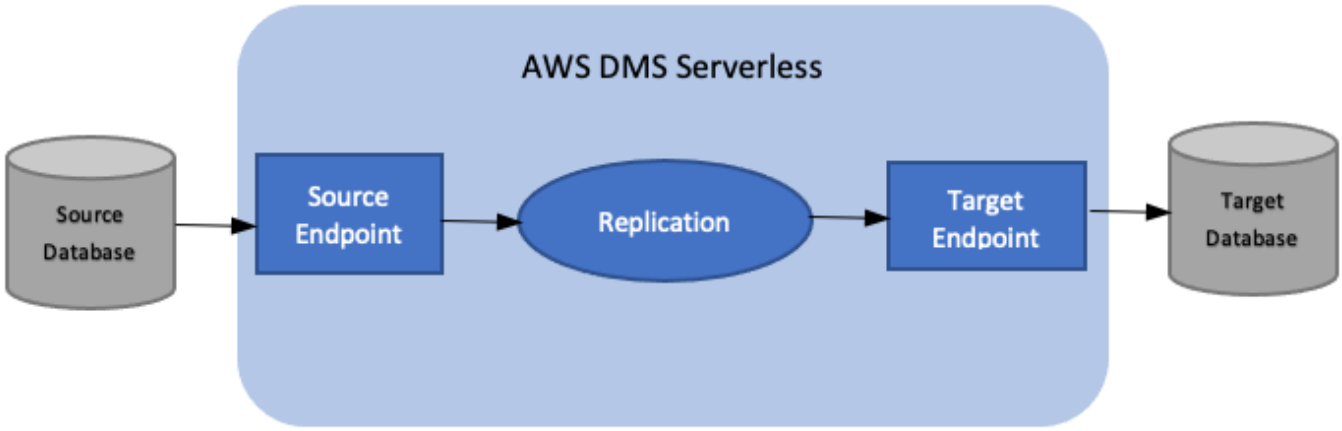
- Tarification AWS Database Migration Service
- AWS Pricing Calculator

Travailler avec AWS DMS Serverless

AWS DMS Serverless est une fonctionnalité qui fournit un provisionnement automatique, une évolutivité, une haute disponibilité intégrée et un modèle de pay-for-use facturation, afin d'accroître l'agilité des opérations et d'optimiser vos coûts. La fonctionnalité sans serveur élimine les tâches de gestion des instances de réplication telles que l'estimation de la capacité, le provisionnement, l'optimisation des coûts et la gestion des versions du moteur de réplication et des correctifs.

Avec AWS DMS Serverless, similaire à la fonctionnalité actuelle de AWS DMS (appelée AWS DMS Standard dans ce document), vous créez des connexions source et cible à l'aide de points de terminaison. Après avoir créé vos points de terminaison sources et cibles, vous créez une configuration de réplication qui inclut les paramètres de configuration pour la réplication donnée. Vous pouvez gérer les réplications en les démarrant, en les arrêtant, en les modifiant ou en les supprimant. Chaque réplication possède des paramètres que vous pouvez configurer selon les exigences de votre migration de base de données. Vous spécifiez ces paramètres à l'aide d'un fichier JSON ou de la AWS DMS section du AWS Management Console. Pour plus d'informations sur les paramètres de réplication, consultez la section Utilisation des AWS DMS points de terminaison. Après le démarrage de la réplication, AWS DMS sans serveur se connecte à la base de données source et collecte les métadonnées de base de données pour analyser la charge de travail de réplication. À l'aide de ces métadonnées, AWS DMS calcule et provisionne la capacité requise et démarre la réplication des données.

Le schéma suivant montre le processus de réplication AWS DMS sans serveur.



> **Note**
>
> AWS DMS Serverless utilise la version du moteur par défaut. Pour en savoir plus sur la version par défaut du moteur, consultez Notes de mise à jour.

Consultez les rubriques suivantes pour en savoir plus sur AWS DMS Serverless.

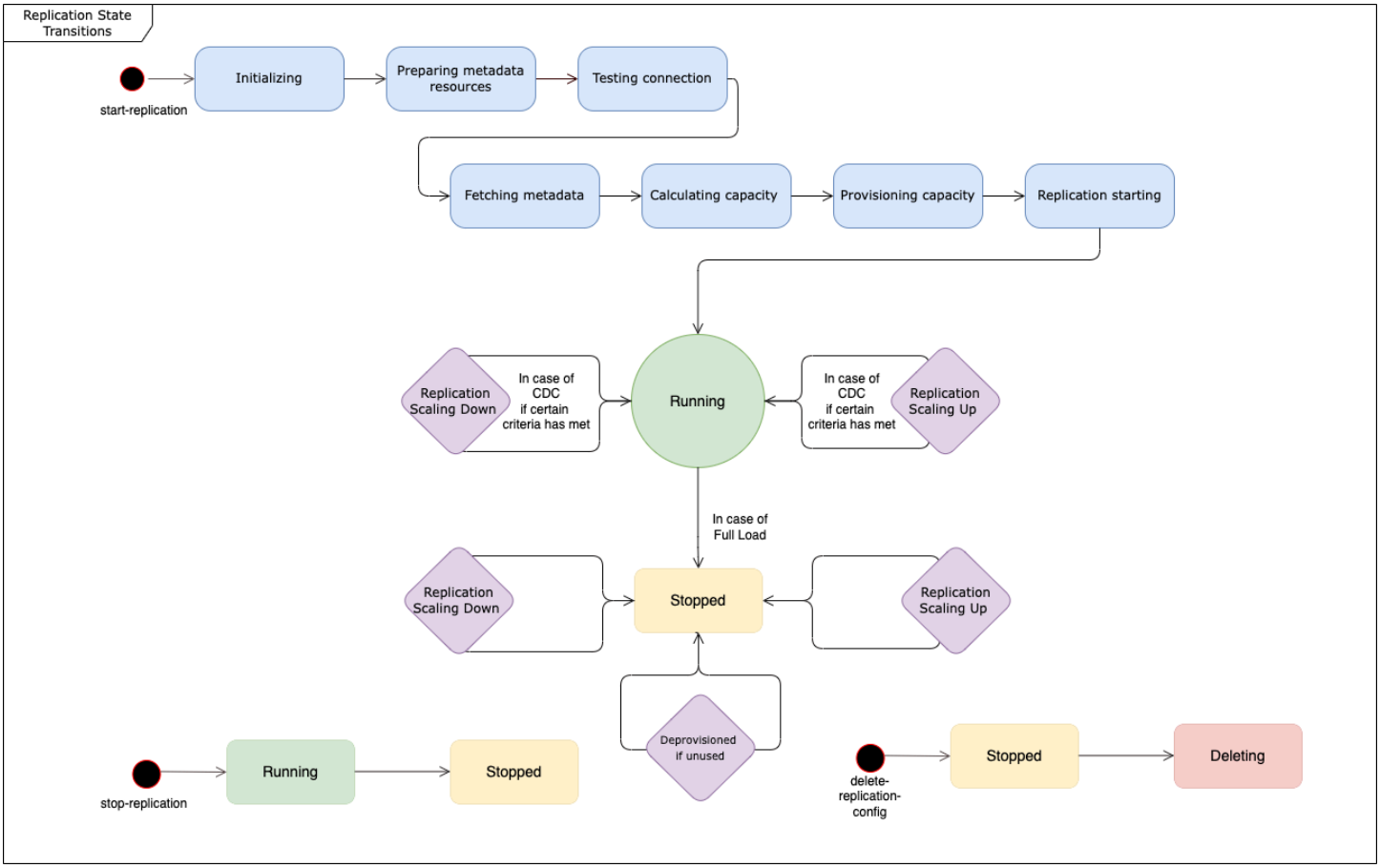
Rubriques

- AWS DMS Composants sans serveur
- AWS DMS Limitations sans serveur

AWS DMS Composants sans serveur

Pour gérer les ressources nécessaires à l'exécution d'une réplication, AWS DMS Serverless possède des états granulaires qui révèlent les différentes actions internes entreprises par le service. Lorsque vous démarrez la réplication, AWS DMS sans serveur calcule la charge de capacité, provisionne la capacité calculée et démarre la réplication des données conformément aux états de réplication suivants.

Le schéma suivant montre les transitions d'état pour une réplication AWS DMS sans serveur.



- Le premier état après le démarrage de la réplication est Initialisation. Dans cet état, tous les paramètres requis sont initialisés.
- Les états suivants incluent Préparation des ressources de métadonnées, Tester la connexion et Récupération de métadonnées. Dans ces états, AWS DMS Serverless se connecte à votre base de données source pour obtenir les informations nécessaires pour prévoir la capacité requise.
  - Lorsque l'état de réplication est Testing Connection, AWS DMS Serverless vérifie que la connexion à vos bases de données source et cible est correctement configurée.
  - L'état de réplication qui suit Tester la connexion est Récupération de métadonnées. Ici, AWS DMS récupère les informations nécessaires au calcul de la capacité.
  - Une fois les AWS DMS informations nécessaires récupérées, l'état suivant est le calcul de la capacité. Ici, le système calcule la taille des ressources sous-jacentes requises pour effectuer la réplication.
- La transition d'état suivant Capacité de calcul est Capacité de provisionnement. Lorsque la réplication est dans cet état, AWS DMS sans serveur initialise les ressources de calcul sous-jacentes.

- L'état de réplication une fois que toutes les ressources ont été correctement provisionnées est Démarrage de la réplication. Dans cet état, AWS DMS Serverless commence la réplication des données.
- L'état final est En cours d'exécution. Dans l'état En cours d'exécution, la réplication des données est en cours.
- Une réplication que vous arrêtez passe à l'état Arrêté. Vous pouvez redémarrer une réplication arrêtée dans les circonstances suivantes :
  - Vous ne pouvez pas redémarrer une réplication que DMS a déprovisionnée.
  - Vous pouvez redémarrer une réplication CDC uniquement ou à chargement complet et une réplication CDC arrêtée à l'aide de cette action. StartReplication Vous ne pouvez pas redémarrer une réplication arrêtée à l'aide de la console.
  - Vous ne pouvez pas redémarrer une réplication arrêtée qui utilise PostgreSQL comme moteur.

Cette rubrique contient les sections suivantes.

- Versions de moteur prises en charge
- Création d'une réplication sans serveur
- Modification des AWS DMS réplications sans serveur
- Configuration de calcul
- Comprendre la mise à l'échelle automatique en mode sans serveur AWS DMS
- Surveillance des AWS DMS réplications sans serveur

Pour AWS DMS Serverless, le panneau de navigation de gauche de la AWS DMS console propose une nouvelle option, les réplications sans serveur. Pour Réplications sans serveur, vous spécifiez Réplications plutôt que des types d'instances de réplication ou des tâches pour définir une réplication. En outre, vous spécifiez les unités de capacité DMS (UCD) maximale et minimale que vous voulez que DMS provisionne pour la réplication. Un DCU correspond à 2 Go de RAM. AWS DMS facture votre compte pour chaque DCU que votre réplication utilise actuellement. Pour plus d'informations sur AWS DMS les tarifs, consultez la section Tarification du AWS Database Migration Service.

AWS DMS provisionne ensuite automatiquement les ressources de réplication en fonction de vos mappages de tables et de la taille prévue de votre charge de travail. Cette unité de capacité est une valeur comprise dans la plage des valeurs d'unité de capacité minimale et maximale que vous spécifiez.

Versions de moteur prises en charge

Avec AWS DMS Serverless, vous n'avez pas besoin de choisir et de gérer les versions du moteur, car le service gère ce paramètre. AWS DMS Serverless prend en charge les sources suivantes :

- Microsoft SQL Server
- Bases de données compatibles PostgreSQL
- Bases de données compatibles MySQL
- MariaDB
- Oracle
- IBM Db2

AWS DMS Serverless prend en charge les cibles suivantes :

- Microsoft SQL Server
- PostgreSQL
- Bases de données compatibles MySQL
- Oracle
- Amazon S3
- Amazon Redshift
- Amazon DynamoDB
- Amazon Kinesis Data Streams
- Amazon Managed Streaming for Apache Kafka
- Amazon OpenSearch Service
- Amazon DocumentDB (compatible avec MongoDB)
- Amazon Neptune

Dans le cadre de AWS DMS Serverless, vous avez accès aux commandes de console qui vous permettent de créer, de configurer, de démarrer et de gérer des réplications AWS DMS sans serveur. Pour exécuter ces commandes à l'aide de la section Réplications sans serveur de la console, vous devez effectuer l'une des opérations suivantes :

- Configurez une nouvelle stratégie AWS Identity and Access Management (IAM) et un nouveau rôle IAM auxquels associer cette politique.
- Utilisez un AWS CloudFormation modèle pour fournir l'accès dont vous avez besoin.

AWS DMS Le mode Serverless nécessite l'existence d'un rôle lié à un service (SLR) dans votre compte. AWS DMS gère la création et l'utilisation de ce rôle. Pour plus d'informations sur la manière de vous assurer que vous disposez du rôle SLR nécessaire, consultez Rôle lié à un service pour AWS DMS sans serveur.

Création d'une réplication sans serveur

Pour créer une réplication sans serveur entre deux AWS DMS points de terminaison existants, procédez comme suit. Pour plus d'informations sur la création de AWS DMS points de terminaison, consultezCréation de points de terminaison source et cible.

Création d'une réplication sans serveur

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Dans le volet de navigation, choisissez Réplications sans serveur, puis Créer une réplication.
3. Sur la page Créer une réplication, spécifiez votre configuration de réplication sans serveur :

Option	Action
Nom	Entrez un nom pour identifier la réplication, tel que **`DMS-replication`** .
Amazon Resource Name (ARN) descriptif : facultatif	Laissez vide .
Point de terminaison de base de données source	Choisissez les points de terminaison existants dans votre compte. Notez que AWS DMS Serverless ne prend en charge qu'un sous-ensemble des types de points de terminaison pris en charge par la AWS DMS norme.
Point de terminaison de base de données cible	Choisissez les points de terminaison existants dans votre compte. Notez que AWS DMS Serverless ne

Option	Action
	prend en charge qu'un sous-ensemble des types de points de terminaison pris en charge par la AWS DMS norme.
Type de réplication	Choisissez un type de réplication en fonction de vos besoins : • Chargement complet : AWS DMS migre uniquement les données existantes. • Capture complète des données à chargement et modification (CDC) : AWS DMS migre les données existantes et les modifications survenues pendant la réplication. • Capture des données de modification (CDC) : migre AWS DMS uniquement les modifications survenant après le démarrage de la réplication.

Dans la section Paramètres, définissez les paramètres requis par la réplication.

Dans la section Mappages de table, configurez le mappage de tables pour définir des règles permettant de sélectionner et de filtrer les données que vous répliquez. Avant de spécifier votre mappage, consultez la section de documentation relative au mappage de type de données pour votre base de données source et cible.

Dans la section Paramètres de calcul, définissez les paramètres suivants. Pour en savoir plus sur les paramètres de configuration de calcul, consultez Configuration de calcul.

Option	Action
VPC	Choisissez un VPC existant.
Groupe de sous-réseaux	Choisissez un groupe de sous-réseaux existant.
Groupes de sécurité VPC	Choisissez par défaut s'il n'est pas déjà sélectionné.

Option	Action
AWS Clé KMS	Laissez (Par défaut) aws/dms sélectionné.
Déploiement	Laissez inchangé.
Zone de disponibilité	Laissez inchangé.
Unités de capacité DMS minimales (UCD) - (facultatif)	Laissez ce champ vide pour utiliser la valeur par défaut de 1 UCD.
Unités de capacité DMS maximales (UCD)	Choisissez 16 UCD.

Laissez les paramètres de maintenance tels quels.

4. Choisissez Créer une réplication.

AWS DMS crée une réplication sans serveur pour effectuer votre migration.

Modification des AWS DMS réplications sans serveur

Pour modifier votre configuration de réplication, utilisez l'action `modify-replication-config`. Vous ne pouvez modifier qu'une configuration de AWS DMS réplication qui est à l'`STOPPED`état`CREATED`, ou`FAILED`.

Pour modifier une configuration de réplication sans serveur à l'aide du AWS Management Console

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Dans le volet de navigation, choisissez Réplications sans serveur.
3. Choisissez la réplication que vous souhaitez modifier. Le tableau suivant décrit les modifications que vous pouvez apporter en fonction de l'état actuel de la réplication.

Paramètre	Description	États autorisés
Nom	Vous pouvez modifier le nom de la réplication. Entrez un nom pour la réplication contenant entre 8 et 16 caractères ASCII imprimables (à l'exception de /, " et @). Ce nom doit être unique pour votre compte dans la région AWS que vous avez sélection née. Vous pouvez choisir d'ajouter quelques détails au nom, comme inclure la AWS région et la tâche que vous effectuez, par exemple :**`west2-mysql2mysql-config1`** .	`ReplicationState` a pour valeur `CREATED`, `STOPPED` ou `FAILED`.
Point de terminaison de base de données source	Choisissez un nouveau point de terminaison source existant en tant que source pour la réplication.	`ReplicationState` a pour valeur `CREATED`, ou `FAILED` quand `ProvisionState` a pour valeur `null`.
Point de terminaison de base de données cible	Choisissez un nouveau point de terminaison cible existant en tant que cible pour la réplication.	`ReplicationState` a pour valeur `CREATED`, ou `FAILED` quand `ProvisionState` a pour valeur `null`.
Type de réplication	Vous pouvez modifier le type d'une réplication sans serveur.	`ReplicationState` a pour valeur `CREATED`, ou `FAILED` quand `ProvisionState` a pour valeur `null`.

Paramètre	Description	États autorisés
Paramètres de réplication	Vous pouvez modifier les paramètres de réplication, y compris le mode de préparation des tables cibles, l'inclusion ou non des colonnes LOB dans la réplication, la taille maximale des objets LOB, la validation et la journalisation. Pour de plus amples informations, veuillez consulter Paramètres de tâche.	`ReplicationState` a pour valeur `CREATED`, `STOPPED` ou `FAILED`.
Mappages de table	Vous pouvez modifier les paramètres de mappage de tables pour une réplication sans serveur, notamment les règles de sélection et les règles de transformation. Pour de plus amples informations, veuillez consulter Mappage de table.	`ReplicationState` a pour valeur `CREATED`, `STOPPED` ou `FAILED`.

Paramètre	Description	États autorisés

Configuration de calcul	Vous pouvez modifier les paramètres de configuration de calcul pour une réplication sans serveur, notamment les paramètres de mise en réseau, les paramètres de mise à l'échelle et les paramètres de maintenance. Pour en savoir plus sur les paramètres de configuration de calcul, consultez Configuration de calcul.	• Vous pouvez modifier les paramètres réseau, de mise à l'échelle et de maintenance suivants quand `ReplicationState` a pour valeur `CREATED`, `STOPPED` ou `FAILED` :
		  • `MinCapacityUnits`
		  • `MaxCapacityUnits`
		  • `MultiAZ`
		  • `PreferredMaintenanceWindow`
		  • `VpcSecurityGroupIds`
		• Vous pouvez modifier les paramètres de mise en réseau et de sécurité suivants quand `ReplicationState` a pour valeur `CREATED`, ou `FAILED` quand `ProvisionState` a pour valeur `null` :
		  • `AvailabilityZone`
		  • `DnsNameServers`

Paramètre	Description	États autorisés
		• `KmsKeyId` • `ReplicationSubnetGroupId`

Configuration de calcul

Vous configurez le provisionnement de la réplication à l'aide de la section de la console ou du paramètre Configuration de calcul. Les champs de l'objet Configuration de calcul incluent les suivants :

Option	Description
MinCapacityUnits	Il s'agit du nombre minimum d'unités de capacité DMS (DCU) qui AWS DMS seront fournies. Il s'agit également de l'UCD minimale à laquelle la mise à l'échelle automatique peut réduire la capacité.
MaxCapacityUnits	Il s'agit des unités de capacité DMS (UCD) maximales qu' AWS DMS peut provisionner, en fonction de la prédiction de capacité de la réplication. Il s'agit également de l'UCD maximale à laquelle la mise à l'échelle automatique peut augmenter la capacité.
KmsKeyId	Clé de chiffrement à utiliser pour chiffrer les informations de connexion et de stockage de réplication. Si vous choisissez (par défaut) aws/dms, AWS DMS utilise la clé KMS par défaut associée à votre compte et. Région AWS Une description et votre numéro de compte sont affichés, ainsi que l'ARN de la clé. Pour plus d'informations sur l'utilisation de la clé de chiffrement, consultez Définition d'une clé de chiffrement et spécification des autorisations AWS KMS. Pour ce didacticiel, laissez (Par défaut) aws/dms sélectionné.

Option	Description
ReplicationSubnetGroupId	Groupe de sous-réseaux de réplication figurant dans le VPC que vous avez sélectionnez, où vous souhaitez que la réplication soit créée. Si la base de données source se trouve dans un VPC, choisissez le groupe de sous-réseaux qui contient la base de données source comme emplacement de réplication. Pour plus d'informations sur les groupes de sous-réseaux de réplication, consultez la page Créer groupe de sous-réseaux de réplication.
VpcSecurityGroupIds	L'instance de réplication est créée dans un VPC. Si la base de données source se trouve dans un VPC, choisissez le groupe de sécurité du VPC qui permet d'accéder à l'instance de base de données où réside la base de données.
PreferredMaintenanceWindow	Ce paramètre définit un intervalle de temps hebdomadaire, au format Universal Coordinated Time (UTC), pendant lequel a lieu la maintenance du système. La valeur par défaut est une fenêtre de 30 minutes sélectionnée au hasard parmi une tranche de 8 heures par période Région AWS, survenant un jour aléatoire de la semaine.
MultiA	La définition de ce paramètre facultatif permet de créer un réplica de secours de la réplication dans une autre zone de disponibilité pour la prise en charge du basculement. Si vous avez l'intention d'utiliser la capture des données de modification (CDC) ou la réplication continue, nous vous recommandons d'activer cette option.

Comprendre la mise à l'échelle automatique en mode sans serveur AWS DMS

Une fois que vous avez configuré une réplication et qu'elle est dans son `RUNNING` état, le AWS DMS service gère la capacité des ressources sous-jacentes à s'adapter à l'évolution des charges de travail. Cette gestion met à l'échelle les ressources de réplication en fonction des paramètres de réplication suivants :

- `MinCapacityUnits`
- `MaxCapacityUnits`

Les réplications augmentent après une période de dépassement d'un seuil maximal d'utilisation et diminuent lorsque l'utilisation de la capacité est inférieure à un seuil minimal d'utilisation de capacité pendant une période prolongée.

Optimisation de la mise à l'échelle automatique en mode sans serveur AWS DMS

Pour ajuster les paramètres de mise à l'échelle automatique de votre réplication, nous vous recommandons de les `MaxCapacityUnits` définir sur la valeur maximale et de laisser AWS DMS gérer le provisionnement des ressources. Il est recommandé de choisir le paramètre de capacité maximale d'UCD le plus élevé afin de tirer le meilleur parti de la mise à l'échelle automatique et de faire face aux pics du volume de transactions. Le calculateur de prix indique le coût mensuel maximum si la réplication utilise en permanence l'UCD maximale. L'UCD maximale ne représente pas le coût réel, car vous ne payez que pour la capacité utilisée.

Si votre réplication n'utilise pas ses ressources à pleine capacité, elle AWS DMS déprovisionnera progressivement les ressources afin de réduire les coûts. Toutefois, étant donné que le provisionnement et le déprovisionnement des ressources prennent du temps, nous vous recommandons de définir votre paramètre `MinCapacityUnits` sur une valeur capable de gérer les pics soudains auxquels vous vous attendez dans votre charge de travail de réplication. Cela empêchera votre réplication d'être sous-provisionnée tout AWS DMS en allouant des ressources pour le niveau de charge de travail plus élevé.

Si vous sous-provisionnez la réplication avec un paramètre de capacité maximale trop bas pour les besoins en données, ou une capacité minimale trop faible pour gérer les pics soudains de votre charge de travail de réplication, il se peut que votre métrique `CapacityUtilization` atteigne régulièrement la valeur maximale. Cela peut entraîner l'échec de la réplication. Si votre réplication échoue en raison d'un sous-provisionnement des ressources, AWS DMS crée un out-of-

memory événement dans vos journaux de réplication. Si le out-of-memory problème est dû à une augmentation soudaine de votre charge de travail de réplication, la réplication sera automatiquement redimensionnée et redémarrera.

Surveillance des AWS DMS réplications sans serveur

AWS fournit plusieurs outils pour surveiller vos réplications AWS DMS sans serveur et répondre aux incidents potentiels :

- AWS DMS métriques de réplication sans serveur
- AWS DMS journaux de réplication sans serveur

AWS DMS métriques de réplication sans serveur

La surveillance de la réplication sans serveur inclut CloudWatch les métriques Amazon pour les statistiques suivantes. Ces statistiques sont groupées par réplication sans serveur individuelle.

Métrique	Unités	Description
CapacityUtilization	Pourcentage	Pourcentage de la mémoire utilisé par la réplication sans serveur
CDC IncomingChanges	Pourcentage	Nombre total d'événements de modification survenus sur un point-in-time site qui attendent d'être appliqués à la cible. Notez que cela est différent de la mesure du taux de modifications de transaction du point de terminaison source. Un nombre élevé pour cette métrique indique généralement qu' AWS DMS il est impossible d'appliquer les modifications capturées en temps opportun, ce qui entraîne une latence cible élevée.
CDC LatencySource	Secondes	Intervalle, en secondes, entre le dernier événement capturé à partir du point de terminaison source et l'horodatage système actuel de l'instance AWS DMS . CDC LatencySource représente le temps de latence entre la source et l'instance de réplication. Un CDC élevé LatencySource signifie que le processus de

Métrique	Unités	Description
		capture des modifications à la source est retardé. Pour identifier le temps de latence dans une réplication en cours, vous pouvez consulter cette métrique conjointe ment avec le CDCLatencyTarget. Si les taux de CDC LatencySource et de CDC LatencyTarget sont élevés, examinez d'LatencySourceabord les CDC. Le CDC LatencySource peut être égal à 0 lorsqu'il n'y a aucun délai de réplication entre la source et la réplication. Le CDC LatencySource peut également atteindre zéro lorsque la réplication tente de lire le prochain événement dans le journal des transactions de la source et qu'il n'y a aucun nouvel événement par rapport à la dernière lecture depuis la source. Lorsque cela se produit, la réplication réinitialise le CDC LatencySource à 0.

Métrique	Unités	Description
CDC LatencyTarget	Secondes	Intervalle, en secondes, entre le premier horodatage d'événement en attente de validation sur la cible et l'horodatage actuel de l'instance AWS DMS . La latence cible est la différence entre l'heure du serveur de l'instance de réplication et le plus ancien identifiant d'événement non confirmé transféré vers un composant cible. En d'autres termes, la latence cible est la différence d'horodatage entre l'instance de réplication et le plus ancien événement appliqué mais non confirmé par le point de terminaison TRG (99 %). Lorsque le CDC LatencyTarget est élevé, cela indique que le processus d'application des événements de changement à la cible est retardé. Pour identifier le temps de latence dans une réplication en cours, vous pouvez consulter cette métrique conjointement avec le CDCLatencySource. Si le taux de CDC LatencyTarget est élevé mais LatencySource pas élevé, vérifiez si : • Aucune clé primaire ni aucun index ne se trouvent dans la cible • Des goulots d'étranglement liés aux ressources se produisent dans la cible ou l'instance de réplication • Les problèmes de réseau se situent entre la réplication et la cible
CDC ThroughputBandwidthTarget	Ko/seconde	Données sortantes transmises pour la cible en Ko par seconde. Le CDC ThroughputBandwidth enregistre les données sortantes transmises sur les points de prélèvement. Si aucun trafic réseau n'est trouvé, la valeur est zéro. Étant donné que CDC ne délivre pas de transactions de longue durée, le trafic réseau peut ne pas être enregistré.
CDC ThroughputRowsSource	Lignes/seconde	Modifications entrantes à partir de la source en lignes par seconde.

Métrique	Unités	Description
CDC Throughput RowsTarget	Lignes/seconde	Modifications sortantes pour la cible en lignes par seconde.
FullLoadThroughput BandwidthTarget	Ko/seconde	Données sortantes transmises à partir d'un chargement complet pour la cible en Ko par seconde.
FullLoadThroughput RowsTarget	Lignes/seconde	Modifications sortantes à partir d'un chargement total depuis la source, exprimées en lignes par seconde.

AWS DMS journaux de réplication sans serveur

Vous pouvez utiliser Amazon CloudWatch pour enregistrer les informations de réplication au cours d'un processus de AWS DMS migration. Vous activez la journalisation lorsque vous sélectionnez les paramètres de réplication.

Les réplications sans serveur chargent les journaux d'état sur votre CloudWatch compte afin de fournir une meilleure visibilité sur la progression de la réplication et de faciliter le dépannage.

AWS DMS télécharge les journaux liés sans serveur vers un groupe de journaux dédié avec le préfixe. `dms-serverless-replication-<your replication config resource ID>` Ce groupe de journaux contient un flux de journaux appelé `dms-serverless-replication-orchestrator-<your replication config resource ID>`. Ce flux de journaux indique l'état de réplication de la réplication et un message associé fournissant des informations supplémentaires sur le travail effectué à ce stade. Pour obtenir des exemples d'entrées de journal, consultez Exemples de journaux de réplication sans serveur ci-dessous.

> **Note**
>
> AWS DMS ne crée ni le groupe de journaux ni le flux tant que vous n'avez pas exécuté la réplication. AWS DMS ne crée pas le groupe de journaux ou le flux si vous créez uniquement la réplication.

Pour afficher les journaux d'une réplication exécutée, procédez comme suit :

1. Ouvrez la AWS DMS console et choisissez Serverless Replications dans le volet de navigation. La boîte de dialogue Réplications sans serveur apparaît.
2. Accédez à la section Configuration et choisissez Afficher les journaux sans serveur dans la colonne Général. Le groupe de CloudWatch journaux s'ouvre.
3. Localisez la section Journaux des tâches de migration et choisissez Afficher CloudWatch les journaux.

Si votre réplication échoue, AWS DMS crée une entrée de journal avec un état de réplication `failed` égal à et un message décrivant la raison de l'échec. Vous devez vérifier vos CloudWatch journaux comme première étape pour résoudre les problèmes liés à un échec de réplication.

> **Note**
>
> Comme avec AWS DMS Classic, vous avez la possibilité d'activer une journalisation plus précise de la progression de la migration des données elle-même, c'est-à-dire des journaux émis par la tâche de réplication sous-jacente. Vous pouvez activer ces journaux dans vos paramètres de réplication en définissant `EnableLogging` dans le champ `Logging` sur `true`, comme dans l'exemple JSON suivant :
>
> ```
> {
>   "Logging": {
>     "EnableLogging": true
>   }
> }
> ```
>
> Si vous activez ces journaux, ils ne commencent à apparaître qu'au cours de la phase `running` de la réplication sans serveur. Ils apparaîtront dans le même groupe de journaux que le flux de journaux précédent, mais figureront sous le nouveau flux de journaux `dms-serverless-serv-res-id-`*`{unique identifier}`*. Consultez la section suivante pour obtenir des informations sur la manière d'interpréter des journaux de réplication sans serveur.

Exemples de journaux de réplication sans serveur

Cette section inclut des exemples d'entrées de journal pour les réplications sans serveur.

Exemple : début de la réplication

Lorsque vous exécutez une réplication sans serveur, AWS DMS crée une entrée de journal similaire à la suivante :

```
{'replication_state':'initializing', 'message': 'Initializing the replication
 workflow.'}
```

Exemple : échec de la réplication

Si l'un des points de terminaison de la réplication n'est pas configuré correctement, AWS DMS crée une entrée de journal similaire à la suivante :

```
{'replication_state':'failed', 'message': 'Test connection failed for endpoint X.',
 'failure_message': 'X'}
```

Si ce message apparaît dans votre journal après un échec, assurez-vous que le point de terminaison spécifié est sain et correctement configuré.

AWS DMS Limitations sans serveur

AWS DMS Le mode Serverless présente les limites suivantes :

- Vous ne pouvez modifier qu'une configuration de AWS DMS réplication qui est à l'`STOPPED`état`CREATED`, ou`FAILED`. Pour obtenir des détails sur les paramètres que vous pouvez modifier dans des conditions spécifiques, consultez Modification des AWS DMS réplications sans serveur.
- Vous ne pouvez supprimer qu'une configuration de AWS DMS réplication qui est à `STOPPED` l'`FAILED`état ou.
- Un stockage statique alloué de 100 Go est disponible pour une réplication. Si la réplication utilise plus de mémoire que cela, en raison d'exigences telles que des transactions de longue durée ou la mise en cache, nous vous recommandons de partitionner votre charge de travail en réplications sans serveur distinctes. Vous pouvez partitionner votre charge de travail par table ou par exigence, par exemple en plaçant toutes les réplications impliquant des LOB dans une réplication sans serveur distincte.
- Contrairement aux instances de réplication, les réplications AWS DMS sans serveur ne disposent pas d'adresse IP publique pour les tâches de gestion. Vous gérez les réplications sans serveur à l'aide de la console.

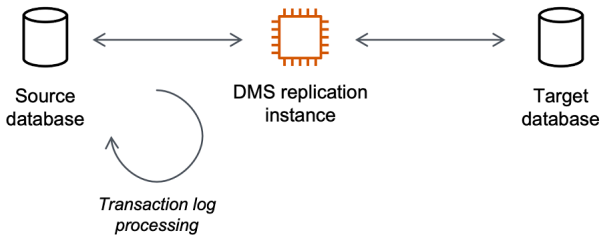
- Cette version de AWS DMS Serverless ne prend pas en charge tous les types de points de terminaison source et cible pris en charge par la AWS DMS norme. Pour afficher la liste des types de moteur pris en charge, consultez AWS DMS Composants sans serveur.
- Les réplications sans serveur doivent accéder aux dépendances à l'aide de points de terminaison de VPC. Vous devez utiliser les points de terminaison de VPC pour accéder aux types de points de terminaison suivants :
  - Amazon Amazon S3
  - Amazon Kinesis
  - AWS Secrets Manager
  - Amazon DynamoDB
  - Amazon Redshift
  - Amazon OpenSearch Service

  Pour en savoir plus sur la configuration de points de terminaison de VPC, consultez Configuration de points de terminaison de VPC en tant que points de terminaison sources et cibles AWS DMS.
- AWS DMS serverless ne prend pas en charge les vues comportant des règles de sélection et de transformation.
- AWS DMS serverless ne prend pas en charge l'utilisation de clés gérées par AWS le client. AWS DMS serverless ne prend en charge que l'utilisation de la clé DMS par défaut. Pour de plus amples informations, veuillez consulter Protection des données dans AWS Database Migration Service.

Utilisation d'une instance AWS DMS de réplication

Lorsque vous créez une instance de AWS DMS réplication, AWS DMS créez-la sur une instance Amazon EC2 dans un cloud privé virtuel (VPC) basé sur le service Amazon VPC. Cette instance de réplication vous permet de réaliser votre migration de base de données. L'utilisation d'une instance de réplication vous permet de bénéficier d'une haute disponibilité et de la prise en charge du basculement avec un déploiement multi-AZ lorsque vous choisissez l'option Multi-AZ.

Dans un déploiement multi-AZ, provisionne et gère AWS DMS automatiquement une réplique de secours synchrone de l'instance de réplication dans une autre zone de disponibilité. L'instance de réplication principale est répliquée de manière synchrone entre les Zones de Disponibilité vers le réplica de secours. Cette approche offre la redondance des données, élimine les figements d'E/S et minimise les pics de latence.



AWS DMS utilise une instance de réplication pour se connecter à votre magasin de données source, lire les données sources et formater les données pour qu'elles soient consommées par le magasin de données cible. Une instance de réplication charge également les données dans le magasin de données cible. La majeure partie de ce traitement se passe dans la mémoire. Cependant, les transactions importantes peuvent nécessiter une mise en mémoire tampon sur disque. Les transactions mises en cache et les fichiers journaux sont également écrits sur le disque.

Vous pouvez créer une instance de AWS DMS réplication dans les AWS régions suivantes.

Nom de la région	Région	Point de terminaison	Protocole	
US East (Ohio)	us-east-2	dms.us-east-2.amazonaws.com	HTTPS	
		dms-fips.us-east-2.amazonaws.com	HTTPS	

Nom de la région	Région	Point de terminaison	Protocole
US East (N. Virginia)	us-east-1	dms.us-east-1.amazonaws.com dms-fips.us-east-1.amazonaws.com	HTTPS HTTPS
USA Ouest (Californie du Nord)	us-west-1	dms.us-west-1.amazonaws.com dms-fips.us-west-1.amazonaws.com	HTTPS HTTPS
USA Ouest (Oregon)	us-west-2	dms.us-west-2.amazonaws.com dms-fips.us-west-2.amazonaws.com	HTTPS HTTPS
Afrique (Le Cap)	af-south-1	dms.af-south-1.amazonaws.com	HTTPS
Asie-Pacifique (Hong Kong)	ap-east-1	dms.ap-east-1.amazonaws.com	HTTPS
Asie-Pacifique (Hyderabad)	ap-south-2	dms.ap-south-2.amazonaws.com	HTTPS
Asie-Pacifique (Jakarta)	ap-southeast-3	dms.ap-southeast-3.amazonaws.com	HTTPS
Asie-Pacifique (Melbourne)	ap-southeast-4	dms.ap-southeast-4.amazonaws.com	HTTPS

Nom de la région	Région	Point de terminaison	Protocole	
Asia Pacific (Mumbai)	ap-south-1	dms.ap-south-1.amazonaws.com	HTTPS	
Asie-Pacifique (Osaka)	ap-northeast-3	dms.ap-northeast-3.amazonaws.com	HTTPS	
Asia Pacific (Seoul)	ap-northeast-2	dms.ap-northeast-2.amazonaws.com	HTTPS	
Asie-Pacifique (Singapour)	ap-southeast-1	dms.ap-southeast-1.amazonaws.com	HTTPS	
Asie-Pacifique (Sydney)	ap-southeast-2	dms.ap-southeast-2.amazonaws.com	HTTPS	
Asie-Pacifique (Tokyo)	ap-northeast-1	dms.ap-northeast-1.amazonaws.com	HTTPS	
Canada (Centre)	ca-central-1	dms.ca-central-1.amazonaws.com	HTTPS	
Canada Ouest (Calgary)	ca-west-1	dms.ca-west-1.amazonaws.com	HTTPS	
Europe (Francfort)	eu-central-1	dms.eu-central-1.amazonaws.com	HTTPS	

Nom de la région	Région	Point de terminaison	Protocole	
Europe (Irlande)	eu-west-1	dms.eu-west-1.amazonaws.com	HTTPS	
Europe (Londres)	eu-west-2	dms.eu-west-2.amazonaws.com	HTTPS	
Europe (Milan)	eu-south-1	dms.eu-south-1.amazonaws.com	HTTPS	
Europe (Paris)	eu-west-3	dms.eu-west-3.amazonaws.com	HTTPS	
Europe (Espagne)	eu-south-2	dms.eu-south-2.amazonaws.com	HTTPS	
Europe (Stockholm)	eu-north-1	dms.eu-north-1.amazonaws.com	HTTPS	
Europe (Zurich)	eu-central-2	dms.eu-central-2.amazonaws.com	HTTPS	
Israël (Tel Aviv)	il-central-1	dms.il-central-1.amazonaws.com	HTTPS	
Moyen-Orient (Bahreïn)	me-south-1	dms.me-south-1.amazonaws.com	HTTPS	
Moyen-Orient (EAU)	me-central-1	dms.me-central-1.amazonaws.com	HTTPS	

Nom de la région	Région	Point de terminaison	Protocole	
Amérique du Sud (São Paulo)	sa-east-1	dms.sa-east-1.amazonaws.com	HTTPS	
AWS GovCloud (USA Est)	us-gov-east-1	dms.us-gov-east-1.amazonaws.com	HTTPS	
AWS GovCloud (US-Ouest)	us-gov-west-1	dms.us-gov-west-1.amazonaws.com	HTTPS	

AWS DMS prend en charge une AWS région spéciale appelée AWS GovCloud (US) , conçue pour permettre aux agences gouvernementales et aux clients américains de transférer des charges de travail sensibles vers le cloud. AWS GovCloud (US) répond aux exigences réglementaires et de conformité spécifiques du gouvernement américain. Pour plus d'informations AWS GovCloud (US), voir Qu'est-ce que AWS GovCloud (États-Unis) ?

Ci-dessous, vous trouverez plus d'informations sur les instances de réplication.

Rubriques

- Choisir l'instance de réplication AWS DMS adaptée à votre migration
- Sélection de la meilleure taille pour une instance de réplication
- Utilisation des versions du moteur de réplication
- Instances de réplication publiques et privées
- Adressage IP et types de réseau
- Configuration d'un réseau pour une instance de réplication
- Définition d'une clé de chiffrement pour une instance de réplication
- Création d'une instance de réplication

- Modification d'une instance de réplication
- redémarrage d'une instance de réplication.
- Supprimez une instance de réplication.
- Utilisation de la fenêtre de maintenance AWS DMS

Choisir l'instance de réplication AWS DMS adaptée à votre migration

AWS DMS crée l'instance de réplication sur une instance Amazon EC2. AWS DMS supporte actuellement les classes d'instances Amazon EC2 T2, T3, C4, C5, C6i, R4, R5 et R6i pour les instances de réplication :

- Les instances T2 sont des instances à performances extensibles qui fournissent un niveau de référence des performances de CPU, avec la possibilité d'émettre en mode rafale au-dessus de ce niveau de référence. Les performances de base et la possibilité d'émettre en rafale sont régies par les crédits UC. Les instances T2 reçoivent des crédits de CPU en continu à un taux défini en fonction de la taille des instances. Elles accumulent des crédits de CPU lorsqu'elles sont inactives et consomment des crédits de CPU lorsqu'elles sont actives.

  Les instances T2 constituent un choix judicieux pour diverses charges de travail à usage général. Celles-ci incluent des microservices, des applications interactives à faible latence, de petites et moyennes bases de données, des bureaux virtuels, des environnements de développement, de génération et intermédiaire, des référentiels de code et des prototypes de produits.
- Les instances T3 constituent le type d'instance à usage général extensible de nouvelle génération. Ce type fournit un niveau de référence de performances de CPU avec la possibilité d'étendre l'utilisation de CPU à tout moment et aussi longtemps que nécessaire. Les instances T3 offrent un équilibre entre les ressources de calcul, de mémoire et de réseau, et sont conçues pour les applications dont l'utilisation de CPU est modérée et qui connaissent des pics d'utilisation temporaires. Les instances T3 accumulent des crédits de CPU quand une charge de travail fonctionne en dessous du seuil de référence. Chaque crédit CPU obtenu donne à l'instance T3 l'occasion de bénéficier des performances d'un cœur de CPU complet pendant une minute en cas de besoin.

  Les instances T3 peuvent émettre en rafales à tout moment et aussi longtemps que nécessaire en mode `unlimited`. Pour plus d'informations sur le mode `unlimited`, consultez Utilisation du mode illimité pour les instances à performances extensibles.

- Les instances C4 sont optimisées pour les charges de travail gourmandes en calcul et offrent des performances élevées à moindre coût, à un faible ratio prix/calcul. Ils permettent d'obtenir des performances de paquets par seconde (PPS) nettement supérieures, de réduire l'instabilité du réseau et de réduire la latence du réseau. AWS DMS peut également nécessiter une utilisation intensive du processeur, en particulier lors de migrations et de réplications hétérogènes telles que la migration d'Oracle vers PostgreSQL. Les instances C4 peuvent être un choix judicieux pour ces situations.
- Les instances C5 constituent le type d'instance de nouvelle génération qui fournit des performances élevées et économiques à un faible rapport prix/calcul pour l'exécution de charges de travail avancées gourmandes en calcul. Cela inclut des charges de travail telles que les serveurs Web hautes performances, le calcul haute performance (HPC), le traitement par lots, la diffusion de publicités, les jeux multijoueurs hautement évolutifs et l'encodage vidéo. Les autres charges de travail pour lesquelles les instances C5 sont adaptées incluent la modélisation scientifique, les analyses distribuées et l'inférence par machine learning et deep learning. Les instances C5 sont disponibles avec un choix de processeurs Intel et AMD.
- Les instances C6i offrent un rapport prix-performance de calcul jusqu'à 15 % supérieur à celui des instances Gen5 comparables pour une grande variété de charges de travail, ainsi qu'un chiffrement de mémoire permanent. Les instances C6 conviennent parfaitement aux charges de travail gourmandes en calcul, telles que le traitement par lots, l'analytique distribuée, le calcul haute performance (HPC), la diffusion de publicités, les jeux multijoueurs hautement évolutifs et l'encodage vidéo.
- Les instances R4 sont optimisées pour la mémoire pour les charges de travail gourmandes en mémoire. Les migrations ou réplications continues de systèmes transactionnels à haut débit utilisant AWS DMS peuvent également consommer de grandes quantités de ressources CPU et de mémoire. Les instances R4 incluent plus de mémoire par vCPU que les types d'instances de générations précédentes.
- Les instances R5 appartiennent à la nouvelle génération de types d'instances à mémoire optimisée pour Amazon EC2. Les instances R5 conviennent aux applications gourmandes en mémoire, telles que les bases de données à hautes performances, les caches en mémoire distribués à l'échelle du web, les bases de données en mémoire de taille moyenne, l'analytique du big data en temps réel et les autres applications d'entreprise. Les migrations ou réplications continues de systèmes de transaction à haut débit AWS DMS peuvent également consommer de grandes quantités de processeur et de mémoire.
- Les instances R6i offrent un rapport prix-performance de calcul jusqu'à 15 % supérieur à celui des instances Gen5 comparables pour une grande variété de charges de travail, ainsi qu'un chiffrement

de mémoire permanent. Les instances R6 sont certifiées SAP et sont idéales pour les charges de travail telles que les bases de données SQL et noSQL, les caches en mémoire distribués à l'échelle du web comme Memcached et Redis, les bases de données en mémoire comme SAP HANA et l'analytique du big data en temps réel comme les clusters Spark et Hadoop.

Chaque instance de réplication a une configuration spécifique de la mémoire et des processeurs virtuels. Le tableau suivant présente la configuration pour chaque type d'instance de réplication. Pour obtenir des informations sur la tarification, consultez la page de tarification du service AWS Database Migration Service.

Types d'instances de réplication à usage général

Type	vCPU	Mémoire (Gio)
dms.t2.micro	1	1
dms.t2.small	1	2
dms.t2.medium	2	4
dms.t2.large	2	8
dms.t3.micro	2	1
dms.t3.small	2	2
dms.t3.medium	2	4
dms.t3.large	2	8

Types d'instances de réplication optimisés pour le calcul

Type	vCPU	Mémoire (Gio)
dms.c4.large	2	3,75
dms.c4.xlarge	4	7,5
dms.c4.2xlarge	8	15

Type	vCPU	Mémoire (Gio)
dms.c4.4xlarge	16	30
dms.c5.large	2	4
dms.c5.xlarge	4	8
dms.c5.2xlarge	8	16
dms.c5.4xlarge	16	32
dms.c5.9xlarge	36	72
dms.c5.12xlarge	48	96
dms.c5.18xlarge	72	144
dms.c5.24xlarge	96	192
dms.c6i.large	2	4
dms.c6i.xlarge	4	8
dms.c6i.2xlarge	8	16
dms.c6i.4xlarge	16	32
dms.c6i.8xlarge	32	64
dms.c6i.12xlarge	48	96
dms.c6i.16xlarge	64	128
dms.c6i.24xlarge	96	192
dms.c6i.32xlarge	128	256

Types d'instances de réplication à mémoire optimisée

Type	vCPU	Mémoire (Gio)
dms.r4.large	2	15,25
dms.r4.xlarge	4	30,5
dms.r4.2xlarge	8	61
dms.r4.4xlarge	16	122
dms.r4.8xlarge	32	244
dms.r5.large	2	16
dms.r5.xlarge	4	32
dms.r5.2xlarge	8	64
dms.r5.4xlarge	16	128
dms.r5.8xlarge	32	256
dms.r5.12xlarge	48	384
dms.r5.16xlarge	64	512
dms.r5.24xlarge	96	768
dms.r6i.large	2	16
dms.r6i.xlarge	4	32
dms.r6i.2xlarge	8	64
dms.r6i.4xlarge	16	128
dms.r6i.8xlarge	32	256
dms.r6i.12xlarge	48	384
dms.r6i.16xlarge	64	512

Type	vCPU	Mémoire (Gio)
dms.r6i.24xlarge	96	768
dms.r6i.32xlarge	128	1 024

Les tableaux ci-dessus répertorient tous les types d'instances de AWS DMS réplication, mais les types disponibles dans votre région peuvent varier. Pour connaître les types d'instances de réplication disponibles dans votre région, vous pouvez exécuter la commande AWS CLI suivante :

```
aws dms describe-orderable-replication-instances --region your_region_name
```

Rubriques

- Choix de la classe d'instances à utiliser
- Utilisation du mode illimité pour les instances à performances extensibles

Choix de la classe d'instances à utiliser

Pour déterminer la classe d'instance de réplication qui vous convient le mieux, examinons le processus de capture des données de modification (CDC) AWS DMS utilisé.

Supposons que vous exécutez une tâche de chargement complet + CDC (chargement en masse + réplication continue). Dans ce cas, la tâche a son propre référentiel SQLite pour stocker les métadonnées et d'autres informations. Avant de AWS DMS démarrer un chargement complet, procédez comme suit :

- AWS DMS commence à capturer les modifications apportées aux tables qu'il migre à partir du journal des transactions du moteur source (nous appelons ces modifications mises en cache). Une fois le chargement complet terminé, ces modifications mises en cache sont collectées et appliquées sur la cible. En fonction du volume des modifications mises en cache, ces modifications peuvent être appliquées directement à partir de la mémoire, où elles sont collectées en premier, jusqu'à un seuil défini. Elles peuvent également être appliquées à partir du disque, où les modifications sont écrites lorsqu'elles ne peuvent pas être conservées en mémoire.
- Une fois les modifications mises en cache appliquées, AWS DMS lance par défaut un processus d'application transactionnel sur l'instance cible.

Au cours de la phase de modifications mises en cache appliquées et de la phase de réplication en cours, AWS DMS utilise deux tampons de flux, un pour les données entrantes et sortantes. AWS DMS utilise également un composant important appelé trieur, qui est une autre mémoire tampon. Voici deux utilisations importantes du composant trieur (qui en a d'autres) :

- Il suit toutes les transactions et s'assure de transférer uniquement les transactions pertinentes vers la mémoire tampon sortante.
- Il s'assure que les transactions sont transférées dans le même ordre de validation que sur la source.

Comme vous pouvez le voir, nous avons trois mémoires tampon importantes dans cette architecture pour CDC dans AWS DMS. Si l'un quelconque de ces tampons connaît une sollicitation importante de mémoire, la migration peut avoir des problèmes de performances susceptibles d'entraîner des échecs.

Lorsque vous connectez d'importantes charges de travail dotées d'un grand nombre de transactions par seconde (TPS) dans cette architecture, la mémoire supplémentaire fournie par les instances R5 et R6i peut s'avérer utile. Vous pouvez utiliser les instances R5 et R6i pour conserver un grand nombre de transactions en mémoire et éviter des problèmes de sollicitation de mémoire au cours des réplications continues.

Utilisation du mode illimité pour les instances à performances extensibles

Une instance à performances extensibles configurée en mode `unlimited`, telle qu'une instance T3, peut maintenir une utilisation de CPU élevée pour toute période donnée, en cas de nécessité. Le prix horaire de l'instance peut couvrir automatiquement tous les pics d'utilisation de CPU. Il le fait si l'utilisation moyenne de CPU de l'instance est égale ou inférieure à la référence sur une période glissante de 24 heures ou pendant la durée de vie de l'instance, si celle-ci est plus courte.

Pour la grande majorité des charges de travail à usage général, les instances configurées comme `unlimited` fournissent des performances suffisantes sans frais supplémentaires. Si l'instance s'exécute avec une utilisation d'UC supérieure pendant une période prolongée, c'est possible moyennant des frais supplémentaires fixes par heure vCPU. Pour en savoir plus sur la tarification des instances T3, consultez « Crédits CPU T3 » dans AWS Database Migration Service.

Pour plus d'informations sur le mode `unlimited` pour les instances T3, consultez Mode illimité pour les instances de performance à capacité extensible dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

> **Important**
>
> Si vous utilisez une instance `dms.t3.micro` dans le cadre de l'offre gratuite AWS et que vous l'utilisez en mode `unlimited`, des frais peuvent s'appliquer. En particulier, des frais peuvent s'appliquer si votre utilisation moyenne sur une période glissante de 24 heures excède l'utilisation de référence de l'instance. Pour plus d'informations, consultez Utilisation de référence dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.
> Les instances T3 sont lancées en mode `unlimited` par défaut. Si l'utilisation moyenne de l'UC sur une période de 24 heures dépasse le niveau de référence, vous devrez payer des frais pour les crédits excédentaires. Dans certains cas, vous pouvez lancer des instances Spot T3 en mode `unlimited` et prévoir de les utiliser immédiatement et pour une courte durée. Si vous effectuez cette opération sans temps d'inactivité pour accumuler des crédits de CPU, vous devrez payer des frais pour les crédits excédentaires. Nous vous recommandons de lancer vos instances Spot T3 en mode standard pour éviter des coûts plus élevés. Pour plus d'informations, consultez Les crédits excédentaires peuvent occasionner des frais, Instances Spot T3 et Mode standard pour les instances de performance à capacité extensible dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Sélection de la meilleure taille pour une instance de réplication

Le choix de l'instance de réplication appropriée dépend de plusieurs facteurs propres à votre cas d'utilisation. Pour mieux comprendre comment les ressources d'instance de réplication sont utilisées, consultez la discussion suivante. Elle couvre le scénario courant d'une tâche de chargement complet + CDC.

Lors d'une tâche de chargement complet, AWS DMS charge les tables individuellement. Par défaut, huit tables sont chargées à la fois. AWS DMS capture les modifications continues apportées à la source pendant une tâche de chargement complet afin que les modifications puissent être appliquées ultérieurement sur le point de terminaison cible. Ces modifications sont mises en cache en mémoire ; si la mémoire disponible est épuisée, les modifications sont mises en cache sur le disque. Lorsqu'une tâche de chargement complet est terminée pour une table, applique AWS DMS immédiatement les modifications mises en cache à la table cible.

Une fois que toutes les modifications mises en cache pour une table ont été appliquées, le point de terminaison cible se trouve dans un état transactionnel cohérent. À ce stade, la cible est synchronisée avec le point de terminaison source par rapport aux dernières modifications mises en

cache. AWS DMS commence ensuite la réplication en cours entre la source et la cible. Pour ce faire, AWS DMS prend les opérations de modification des journaux de transactions source et les applique à la cible de manière cohérente sur le plan des transactions. (Ce processus suppose que l'option Appliquer optimisée par lots n'est pas sélectionnée). AWS DMS diffuse les modifications en cours via la mémoire de l'instance de réplication, si possible. Sinon, AWS DMS écrit les modifications sur le disque de l'instance de réplication jusqu'à ce qu'elles puissent être appliquées à la cible.

Vous pouvez contrôler la façon dont l'instance de réplication gère le traitement des modifications et la façon dont la mémoire est utilisée dans ce processus. Pour plus d'informations sur la manière d'ajuster le traitement des modifications, consultez Paramètres de réglage du traitement des modifications.

Facteurs à prendre en compte

La mémoire et l'espace disque sont des facteurs clés pour la sélection d'une instance de réplication appropriée à votre cas d'utilisation. Vous trouverez ci-dessous une présentation des caractéristiques des cas d'utilisation à analyser pour choisir une instance de réplication.

- Taille des tables et de la base de données

  Le volume de données permet de déterminer la configuration des tâches afin d'optimiser les performances de chargement complet. Par exemple, pour deux schémas de 1 To, vous pouvez partitionner les tables en quatre tâches de 500 Go et les exécuter en parallèle. Le parallélisme possible dépend de la ressource CPU disponible dans l'instance de réplication. C'est pourquoi il est judicieux de connaître la taille de votre base de données et de vos tables afin d'optimiser les performances de chargement complet. Cela permet de déterminer le nombre de tâches que vous pouvez éventuellement effectuer.

- Objets volumineux

  Les types de données présents dans l'étendue de migration peuvent affecter les performances. Les objets volumineux (LOB) ont notamment un impact sur les performances et la consommation de mémoire. Pour migrer une valeur LOB, AWS DMS exécute un processus en deux étapes. Tout d'abord, AWS DMS insère la ligne dans la cible sans la valeur LOB. Ensuite, AWS DMS met à jour la ligne avec la valeur LOB. Cela a un impact sur la mémoire, il est donc important d'identifier les colonnes LOB dans la source et d'analyser leur taille.

- Fréquence de chargement et taille des transactions

La fréquence de chargement et le nombre de transactions par seconde (TPS) influencent l'utilisation de la mémoire. Un nombre élevé du TPS ou des activités de langage de manipulation de données (DML) entraîne une utilisation importante de mémoire. Cela se produit parce que DMS met en cache les modifications jusqu'à ce qu'elles soient appliquées à la cible. Pendant la CDC, cela entraîne un échange (écriture sur le disque physique en raison d'un débordement de mémoire), qui génère de la latence.

- Clés de table et intégrité référentielle

  Les informations relatives aux clés de la table déterminent le mode CDC (application par lots ou application transactionnelle) que vous utilisez pour migrer les données. En général, l'application transactionnelle est plus lente que l'application par lots. Pour les transactions de longue durée, il peut y avoir de nombreuses modifications à migrer. Lorsque vous utilisez l'application transactionnelle, le stockage des modifications AWS DMS peut nécessiter plus de mémoire que l'application par lots. Si vous migrez des tables sans clés primaires, l'application par lots échoue et la tâche DMS passe en mode d'application transactionnelle. Lorsque l'intégrité référentielle est active entre les tables pendant le CDC, AWS DMS utilise l'application transactionnelle par défaut. Pour plus d'informations sur l'application par lots par rapport à l'application transactionnelle, consultez Comment puis-je utiliser la fonctionnalité d'application par lots DMS pour améliorer les performances de réplication CDC ?.

Utilisez ces métriques pour déterminer si l'instance de réplication doit être optimisée pour le calcul ou pour la mémoire.

Problèmes courants

Vous pouvez être confronté aux problèmes courants suivants qui entraînent une contention des ressources sur l'instance de réplication lors de la migration. Pour en savoir plus sur les métriques de l'instance de réplication, consultez Métriques des instances de réplication.

- Si la mémoire d'une instance de réplication devient insuffisante, cela entraîne l'écriture de données sur le disque. La lecture depuis le disque peut entraîner une latence, que vous pouvez éviter en dimensionnant l'instance de réplication avec suffisamment de mémoire.
- La taille de disque attribuée à l'instance de réplication peut être inférieure à celle requise. La taille du disque est utilisée quand les données en mémoire débordent ; elle est également utilisée pour stocker les journaux de tâches. Le nombre maximal d'IOPS en dépend également.

- L'exécution de plusieurs tâches ou de tâches présentant un parallélisme élevé affecte la consommation de CPU de l'instance de réplication. Cela ralentit le traitement des tâches et génère de la latence.

Bonnes pratiques

Tenez compte de ces deux bonnes pratiques les plus courantes lors du dimensionnement d'une instance de réplication. Pour plus d'informations, consultez Bonnes pratiques pour AWS Database Migration Service.

1. Dimensionnez votre charge de travail et déterminez si elle est gourmande en ressources informatiques ou en mémoire. Sur cette base, vous pouvez déterminer la classe et la taille de l'instance de réplication :
   - AWS DMS traite les LOB en mémoire. Cette opération nécessite une quantité importante de mémoire.
   - Le nombre de tâches et le nombre de threads ont un impact sur la consommation de CPU. Évitez d'en utiliser plus de huit `MaxFullLoadSubTasks` pendant l'opération de chargement complet.
2. Augmentez l'espace disque attribué à l'instance de réplication lorsque la charge de travail est élevée pendant le chargement complet. Cela permet à l'instance de réplication d'utiliser le nombre maximal d'IOPS qui lui est attribué.

Les directives précédentes ne couvrent pas tous les scénarios possibles. Il est important de prendre en compte les spécificités de votre cas d'utilisation particulier lorsque vous déterminez la taille de votre instance de réplication.

Les tests précédents montrent que le CPU et la mémoire varient avec différentes charges de travail. En particulier, les objets LOB affectent la mémoire, et le nombre de tâches ou le parallélisme affectent le CPU. Une fois que votre migration est en cours d'exécution, surveillez le CPU, la mémoire libérable, le stockage disponible et les IOPS de votre instance de réplication. En fonction des données que vous collectez, vous pouvez dimensionner votre instance de réplication à la hausse ou à la baisse, selon les besoins.

Utilisation des versions du moteur de réplication

Le moteur de réplication est le AWS DMS logiciel principal qui s'exécute sur votre instance de réplication et exécute les tâches de migration que vous spécifiez. AWS publie régulièrement de nouvelles versions du logiciel du moteur de AWS DMS réplication, avec de nouvelles fonctionnalités et des améliorations de performances. Chaque version du moteur de réplication a son propre numéro, pour la distinguer des autres versions.

Lorsque vous lancez une nouvelle instance de réplication, celle-ci exécute la dernière version AWS DMS du moteur, sauf indication contraire de votre part. Pour plus d'informations, consultez Utilisation d'une instance AWS DMS de réplication.

Si une instance de réplication est actuellement en cours d'exécution, vous pouvez la mettre à niveau vers une version du moteur plus récente. (AWS DMS ne prend pas en charge les rétrogradations de version du moteur.) Pour de plus amples informations sur les versions du moteur de réplication, veuillez consulter AWS Notes de mise à jour du DMS.

Mise à niveau de la version du moteur à l'aide de la console

Vous pouvez mettre à niveau une instance AWS DMS de réplication à l'aide du AWS Management Console.

Pour mettre à niveau une instance de réplication à l'aide de la console

1. Ouvrez la AWS DMS console à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Dans le volet de navigation, sélectionnez Instances de réplication.
3. Choisissez votre moteur de réplication, puis choisissez Modifier.
4. Pour Version du moteur, choisissez le numéro de version de votre choix, puis choisissez Modifier.

> **Note**
>
> Nous vous recommandons d'arrêter toutes les tâches avant de mettre à niveau l'instance de réplication. Si vous n'arrêtez pas la tâche, elle l' AWS DMS arrêtera automatiquement avant la mise à niveau. Si vous arrêtez la tâche manuellement, vous devrez la démarrer manuellement une fois la mise à niveau terminée. La mise à niveau de l'instance de réplication prend plusieurs minutes. Lorsque l'instance est prête, son statut passe à available.

Mise à niveau de la version du moteur à l'aide du AWS CLI

Vous pouvez mettre à niveau une instance de AWS DMS réplication à l'aide du AWS CLI, comme suit.

Pour mettre à niveau une instance de réplication à l'aide du AWS CLI

1. Déterminez le nom Amazon Resource Name (ARN) de votre instance de réplication à l'aide de la commande suivante.

   ```
   aws dms describe-replication-instances \
   --query "ReplicationInstances[*].
   [ReplicationInstanceIdentifier,ReplicationInstanceArn,ReplicationInstanceClass]"
   ```

   Dans la sortie, prenez note du nom ARN pour l'instance de réplication que vous souhaitez mettre à niveau ; par exemple : `arn:aws:dms:us-east-1:123456789012:rep:6EFQQO6U6EDPRCPKLNPL2SCEEY`

2. Déterminez quelles versions d'instance de réplication sont disponibles à l'aide de la commande suivante.

   ```
   aws dms describe-orderable-replication-instances \
   --query "OrderableReplicationInstances[*].[ReplicationInstanceClass,EngineVersion]"
   ```

   Dans la sortie, prenez note du ou des numéros de version du moteur disponibles pour votre classe d'instances de réplication. Vous devriez voir ces informations dans la sortie à partir de l'étape 1.

3. Mettez à niveau l'instance de réplication à l'aide de la commande suivante.

   ```
   aws dms modify-replication-instance \
   --replication-instance-arn arn \
   --engine-version n.n.n
   ```

   Remplacez *arn* dans ce qui précède par le nom ARN de l'instance de réplication réelle obtenu à l'étape précédente.

   Remplacez *n.n.n* par le numéro de version du moteur de votre choix ; par exemple : `3.4.5`

> **Note**
>
> La mise à niveau de l'instance de réplication prend plusieurs minutes. Vous pouvez afficher le statut de l'instance de réplication à l'aide de la commande suivante.
>
> ```
> aws dms describe-replication-instances \
> --query "ReplicationInstances[*].
> [ReplicationInstanceIdentifier,ReplicationInstanceStatus]"
> ```
>
> Lorsque l'instance de réplication est prête, son statut passe à available.

Instances de réplication publiques et privées

Vous pouvez spécifier si une instance de réplication dispose d'une adresse IP publique ou privée que l'instance utilise pour se connecter aux bases de données source et cible.

Une instance de réplication privée a une adresse IP privée à laquelle vous ne pouvez pas accéder en dehors du réseau de réplication. Vous utilisez une instance privée lorsque les bases de données source et cible se trouvent sur le même réseau connecté au cloud privé virtuel (VPC) de l'instance de réplication. Le réseau peut être connecté au VPC à l'aide d'un réseau privé virtuel (VPN) ou d'un AWS Direct Connect peering VPC.

Une connexion d'appairage de VPC est une connexion de mise en réseau entre deux VPC. Il permet le routage en utilisant les adresses IP privées de chaque VPC comme si elles se trouvaient sur le même réseau. Pour de plus amples informations sur l'appairage de VPC, veuillez consulter la section Appairage de VPC dans le Guide de l'utilisateur Amazon VPC.

Une instance de réplication publique peut utiliser le groupe de sécurité de VPC de l'instance de réplication et l'adresse IP publique de l'instance de réplication ou l'adresse IP publique de la passerelle NAT. Ces connexions forment un réseau que vous utilisez pour la migration des données.

Adressage IP et types de réseau

AWS DMS crée toujours votre instance de réplication dans un Amazon Virtual Private Cloud (VPC). Lorsque vous créez votre VPC, vous pouvez déterminer l'adressage IP à utiliser : IPv4 ou IPv6, ou les deux. Ensuite, lorsque vous créez ou modifiez une instance de réplication, vous pouvez spécifier l'utilisation d'un protocole d'adresse IPv4 ou d'un protocole d'adresse IPv6 en mode double pile.

Adresses IPv4

Lorsque vous créez un VPC, vous devez spécifier une plage d'adresses IPv4 pour le VPC sous la forme d'un bloc de routage inter-domaines sans classe (CIDR), tel que 10.0.0.0/16. Un groupe de sous-réseaux définit la plage d'adresses IP de ce bloc CIDR. Ces adresses IP peuvent être privées ou publiques.

Une adresse IPv4 privée est une adresse IP qui ne peut pas être atteinte via Internet. Vous pouvez utiliser des adresses IPv4 privées pour la communication entre votre instance de réplication et d'autres ressources, telles que les instances Amazon EC2, dans le même VPC. Chaque instance de réplication dispose d'une adresse IP privée pour la communication dans le VPC.

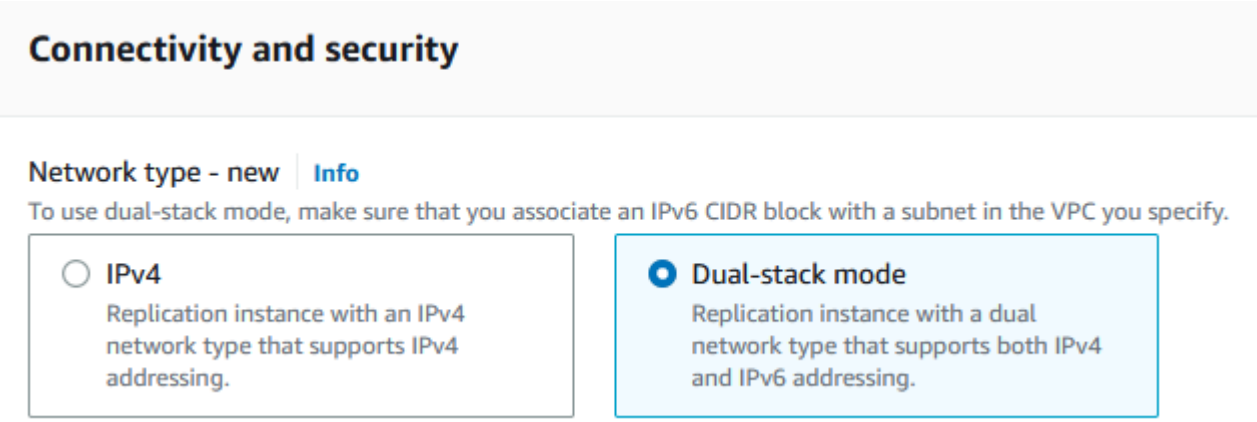
Une adresse IP publique est une adresse IPv4, qui est accessible depuis Internet. Vous pouvez utiliser des adresses publiques pour la communication entre votre instance de réplication et les ressources sur Internet. Vous contrôlez si votre instance de réplication reçoit une adresse IP publique.

Mode double pile et adresses IPv6

Si vous possédez des ressources qui doivent communiquer avec votre instance de réplication via IPv6, utilisez le mode double pile. Pour utiliser le mode double pile, assurez-vous que chaque sous-réseau du groupe de sous-réseaux que vous associez à l'instance de réplication est associé à un bloc CIDR IPv6. Vous pouvez créer un nouveau groupe de sous-réseaux de réplication ou modifier un groupe de sous-réseaux de réplication existant pour répondre à cette exigence. Chaque adresse IPv6 est unique au niveau mondial. Le bloc CIDR IPv6 de votre VPC est automatiquement attribué à partir du groupe d'adresses IPv6 d'Amazon. Vous ne pouvez pas choisir la plage vous-même.

DMS désactive l'accès à la passerelle Internet pour les points de terminaison IPv6 des instances de réplication privées en mode double pile. DMS fait cela pour s'assurer que vos points de terminaison IPv6 sont privés et sont uniquement accessibles depuis votre VPC.

Vous pouvez utiliser la AWS DMS console pour créer ou modifier une instance de réplication et spécifier le mode double pile dans la section Type de réseau. L'image suivante présente la section Network type (Type de réseau) dans la console.



Références

- Pour plus d'informations sur les adresses IPv4 et IPv6, consultez Adressage IP dans le Guide de l'utilisateur Amazon VPC.
- Pour plus d'informations sur la création d'une instance de réplication à l'aide du mode double pile, consultez Création d'une instance de réplication.
- Pour plus d'informations sur la modification d'une instance de réplication, consultez Modification d'une instance de réplication.

Configuration d'un réseau pour une instance de réplication

AWS DMS crée toujours l'instance de réplication dans un VPC basé sur Amazon VPC. Vous spécifiez le VPC dans lequel votre instance de réplication est située. Vous pouvez utiliser votre VPC par défaut pour votre compte et votre AWS région, ou vous pouvez créer un nouveau VPC.

Veillez à ce que l'interface réseau Elastic allouée pour le VPC de votre instance de réplication soit associée à un groupe de sécurité. Veillez également à ce que les règles de ce groupe de sécurité autorisent tout le trafic sur tous les ports à quitter (sortir) le VPC. Cette approche permet la communication de l'instance de réplication vers vos points de terminaison de base de données source et cible, si les bonnes règles de trafic sortant sont activées sur les points de terminaison. Nous recommandons l'utilisation des paramètres par défaut pour les points de terminaison, qui autorisent le trafic sortant sur tous les ports vers toutes les adresses.

Les points de terminaison source et cible accèdent à l'instance de réplication située à l'intérieur du VPC en se connectant au VPC ou en étant à l'intérieur du VPC. Les points de terminaison de base de données doivent inclure des listes ACL réseau et des règles de groupe de sécurité (le cas échéant)

qui autorisent l'accès entrant à partir de l'instance de réplication. La façon dont vous configurez cela dépend de la configuration réseau que vous utilisez. Vous pouvez utiliser le groupe de sécurité de VPC de l'instance de réplication, l'adresse IP privée ou publique de l'instance de réplication ou l'adresse IP publique de la passerelle NAT. Ces connexions forment un réseau que vous utilisez pour la migration des données.

> **Note**
>
> Comme une adresse IP peut changer à la suite de modifications apportées à l'infrastructure sous-jacente, nous vous recommandons d'utiliser la plage d'adresses CIDR d'un VPC ou de router le trafic sortant de votre instance de réplication via une adresse IP Elastic associée à une passerelle NAT. Pour plus d'informations sur la création d'un VPC, y compris d'un bloc CIDR, consultez Utilisation des VPC et des sous-réseaux dans le Guide de l'utilisateur Amazon Virtual Private Cloud. Pour plus d'informations sur les adresses IP Elastic, consultez Adresses IP Elastic dans le Guide de l'utilisateur Amazon Elastic Compute Cloud.

Configurations réseau pour la migration de base de données

Vous pouvez utiliser différentes configurations réseau avec AWS Database Migration Service. Voici quelques configurations courantes pour un réseau utilisé pour la migration de base de données.

Rubriques

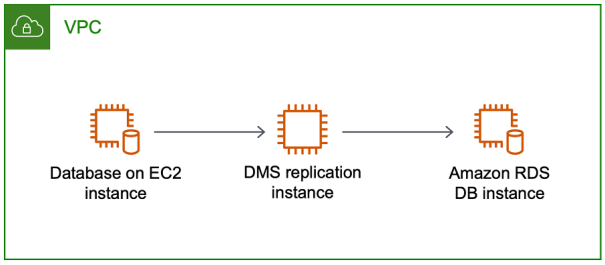
- Configuration avec tous les composants de migration de base de données dans un VPC
- Configuration avec plusieurs VPC
- Configuration avec des VPC partagés
- Configuration d'un réseau vers un VPC à l'aide AWS Direct Connect d'un VPN
- Configuration d'un réseau à un VPC avec Internet
- Configuration avec une instance de base de données RDS ne figurant pas dans un VPC vers une instance de base de données dans un VPC en utilisant ClassicLink

Dans la mesure du possible, nous vous recommandons de créer une instance de réplication DMS dans la même région que votre point de terminaison cible, et dans le même VPC ou sous-réseau que votre point de terminaison cible.

Configuration avec tous les composants de migration de base de données dans un VPC

Le réseau le plus simple pour la migration de base de données consiste à rassembler le point de terminaison source, l'instance de réplication et le point de terminaison cible dans le même VPC. Cette configuration convient si vos points de terminaison sources et cibles se trouvent sur une instance de base de données Amazon RDS ou une instance Amazon EC2.

L'illustration suivante présente une configuration dans laquelle une base de données sur une instance Amazon EC2 se connecte à l'instance de réplication et les données sont migrées vers une instance de base de données Amazon RDS.



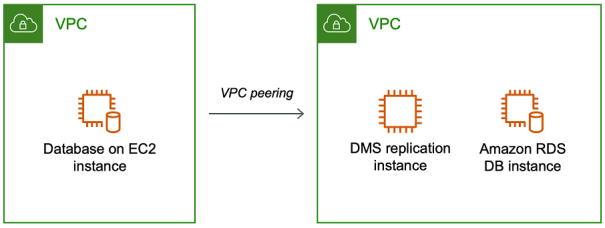
Le groupe de sécurité du VPC utilisé dans cette configuration doit autoriser le trafic entrant sur le port de base de données à partir de l'instance de réplication. Vous pouvez effectuer cette opération de plusieurs manières. Vous pouvez vous assurer que le groupe de sécurité utilisé par l'instance de réplication ait accès aux points de terminaison. Vous pouvez également autoriser la plage CIDR du VPC, l'adresse IP Elastic de la passerelle NAT ou l'adresse IP privée de l'instance de réplication si vous en utilisez une. Toutefois, nous vous déconseillons d'utiliser l'adresse IP privée de l'instance de réplication, car elle peut interrompre la réplication en cas de modification de l'adresse IP de réplication.

Configuration avec plusieurs VPC

Si le point de terminaison source et les points de terminaison cibles se trouvent dans des VPC distincts, vous pouvez créer votre instance de réplication dans l'un de ces VPC. Vous pouvez ensuite lier les deux VPC en utilisant l'appairage de VPC.

Une connexion d'appairage de VPC est une connexion réseau entre deux VPC qui permet le routage en utilisant les adresses IP privées de chaque VPC comme s'ils se trouvaient sur le même réseau. Vous pouvez créer une connexion d'appairage VPC entre vos propres VPC, avec un VPC d'un autre compte ou avec un VPC d'une autre AWS région. AWS Pour de plus amples informations sur l'appairage de VPC, veuillez consulter la section Appairage de VPC dans le Guide de l'utilisateur Amazon VPC.

L'illustration suivante présente un exemple de configuration utilisant l'appairage de VPC. Ici, la base de données source sur une instance Amazon EC2 dans un VPC se connecte par appairage de VPC à un VPC. Ce VPC contient l'instance de réplication et la base de données cible sur une instance de base de données Amazon RDS.



Pour mettre en œuvre l'appairage de VPC, suivez les instructions figurant dans Utilisation de connexions d'appairage de VPC, dans la documentation Amazon Virtual Private Cloud – Appairage de VPC. Assurez-vous que la table de routage d'un VPC contient le bloc CIDR de l'autre. Par exemple, si le VPC A utilise la destination 10.0.0.0/16 et que le VPC B utilise la destination 172.31.0.0, la table de routage du VPC A doit contenir 172.31.0.0 et la table de routage du VPC B doit contenir 10.0.0.0/16. Pour en savoir plus, consultez Mise à jour de vos tables de routage pour une connexion d'appairage de VPC dans la documentation Amazon Virtual Private Cloud – Appairage de VPC.

Les groupes de sécurité de VPC utilisés dans cette configuration doivent autoriser le trafic entrant sur le port de base de données à partir de l'instance de réplication, ou doivent autoriser le trafic entrant sur le bloc CIDR du VPC en cours d'appairage.

Configuration avec des VPC partagés

AWS DMS traite les sous-réseaux partagés avec un compte client participant au sein d'une organisation comme les sous-réseaux ordinaires d'un même compte. Vous trouverez ci-dessous une description de la gestion des AWS DMS VPC, des sous-réseaux et de la manière dont vous pouvez utiliser les VPC partagés.

Vous pouvez configurer votre configuration réseau pour qu'elle fonctionne dans des sous-réseaux ou des VPC personnalisés en créant des objets `ReplicationSubnetGroup`. Lorsque vous créez un `ReplicationSubnetGroup`, vous pouvez choisir de spécifier des sous-réseaux d'un VPC spécifique dans votre compte. La liste des sous-réseaux que vous spécifiez doit inclure au moins deux sous-réseaux situés dans des zones de disponibilité distinctes, et tous les sous-réseaux doivent se trouver dans le même VPC. Lors de la création d'un`ReplicationSubnetGroup`, les clients spécifient uniquement des sous-réseaux. AWS DMS déterminera le VPC en votre nom, car chaque sous-réseau est lié à exactement un VPC.

Lorsque vous créez un AWS DMS `ReplicationInstance` ou un AWS DMS `ReplicationConfig`, vous pouvez choisir de spécifier un `ReplicationSubnetGroup` et/ou un groupe de sécurité VPC dans lequel la `ReplicationInstance` réplication sans serveur fonctionne. S'il n'est pas spécifié, AWS DMS choisit le client par défaut `ReplicationSubnetGroup` (qui est AWS DMS créé en votre nom s'il n'est pas spécifié pour tous les sous-réseaux du VPC par défaut) et le groupe de sécurité VPC par défaut.

Vous pouvez choisir d'exécuter vos migrations dans une zone de disponibilité que vous spécifiez ou dans l'une des zones de disponibilité de votre élément `ReplicationSubnetGroup`. Lorsque vous AWS DMS tentez de créer une instance de réplication ou de démarrer une réplication sans serveur, les zones de disponibilité de vos sous-réseaux sont converties en zones de disponibilité dans le compte de service principal, afin de garantir le lancement des instances dans la bonne zone de disponibilité, même si les mappages de zones de disponibilité ne sont pas identiques entre les deux comptes.

Si vous utilisez un VPC partagé, vous devez vous assurer de créer des objets `ReplicationSubnetGroup` qui correspondent aux sous-réseaux que vous souhaitez utiliser à partir d'un VPC partagé. Lorsque vous créez un élément `ReplicationInstance` ou `ReplicationConfig`, vous devez spécifier un élément `ReplicationSubnetGroup` pour le VPC partagé et spécifier un groupe de sécurité de VPC que vous avez créé pour votre VPC partagé avec votre demande de création.

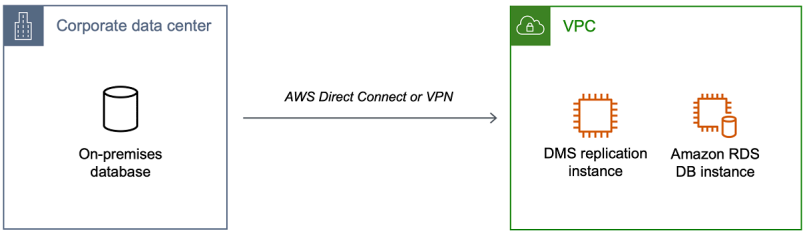
Notez ce qui suit à propos de l'utilisation d'un VPC partagé :

- Le propriétaire du VPC ne peut pas partager une ressource avec un participant, mais celui-ci peut créer une ressource de service dans le sous-réseau du propriétaire.
- Le propriétaire du VPC ne peut pas accéder à une ressource (telle qu'une instance de réplication) créée par le participant, car toutes les ressources sont spécifiques au compte. Toutefois, tant que vous créez l'instance de réplication dans le VPC partagé, elle peut accéder aux ressources du VPC quel que soit le compte propriétaire, à condition que le point de terminaison ou la tâche de réplication dispose des autorisations appropriées.
- Les ressources étant spécifiques à un compte, les autres participants ne peuvent pas accéder aux ressources appartenant à d'autres comptes. Vous ne pouvez accorder aucune autorisation à d'autres comptes pour leur permettre d'accéder aux ressources créées dans le VPC partagé avec votre compte.

Configuration d'un réseau vers un VPC à l'aide AWS Direct Connect d'un VPN

Les réseaux distants peuvent se connecter à un VPC à l'aide de plusieurs options telles que AWS Direct Connect ou une connexion VPN logicielle ou matérielle. Ces options sont souvent utilisées pour intégrer des services sur site existants, comme la surveillance, l'authentification, la sécurité, les données ou d'autres systèmes, en étendant un réseau interne dans le cloud AWS . Grâce à ce type d'extension de réseau, vous pouvez en toute transparence vous connecter à des ressources hébergées par AWS, telles qu'un VPC.

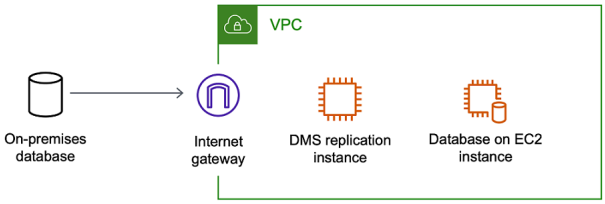
L'illustration suivante présente une configuration dans laquelle le point de terminaison source est une base de données sur site dans un centre de données d'entreprise. Il est connecté via AWS Direct Connect ou un VPN à un VPC qui contient l'instance de réplication et une base de données cible sur une instance de base de données Amazon RDS.



Dans cette configuration, le groupe de sécurité de VPC doit inclure une règle de routage qui envoie à un hôte le trafic destiné à la plage CIDR d'un VPC ou à une adresse IP spécifique. Cet hôte doit être en mesure d'acheminer le trafic du VPC vers le VPN sur site. Dans ce cas, l'hôte NAT inclut ses propres paramètres de groupe de sécurité. Ces paramètres doivent autoriser le trafic provenant de la plage CIDR de VPC de l'instance de réplication, d'une adresse IP privée ou du groupe de sécurité dans l'instance NAT. Toutefois, nous vous déconseillons d'utiliser l'adresse IP privée de l'instance de réplication, car elle peut interrompre la réplication en cas de modification de l'adresse IP de réplication.

Configuration d'un réseau à un VPC avec Internet

Si vous n'utilisez pas de VPN ou AWS Direct Connect si vous ne vous connectez pas à AWS des ressources, vous pouvez utiliser Internet pour migrer votre base de données. Dans ce cas, vous pouvez effectuer une migration vers une instance Amazon EC2 ou vers une instance de base de données Amazon RDS. Cette configuration implique une instance de réplication publique dans un VPC avec une passerelle Internet qui contient le point de terminaison cible et l'instance de réplication.



Pour ajouter une passerelle Internet à votre VPC, veuillez consulter Attachement d'une passerelle Internet dans le Guide de l'utilisateur Amazon VPC.

La table de routage du VPC doit inclure des règles de routage qui envoient par défaut le trafic non destiné au VPC vers la passerelle Internet. Dans cette configuration, la connexion au point de terminaison semble venir de l'adresse IP publique de l'instance de réplication et non pas de l'adresse IP privée. Pour plus d'informations, consultez Tables de routage de VPC dans le Guide de l'utilisateur Amazon VPC.

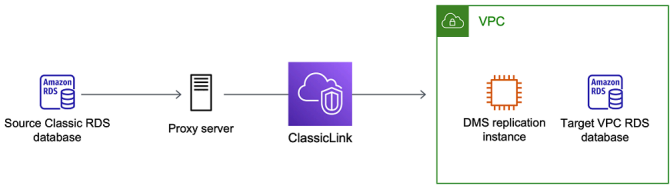
Configuration avec une instance de base de données RDS ne figurant pas dans un VPC vers une instance de base de données dans un VPC en utilisant ClassicLink

> Nous retirons EC2-Classic le 15 août 2022. Nous vous recommandons de migrer d'EC2-Classic vers un VPC. Pour plus d'informations, consultez Migrer d'EC2-Classic vers un VPC dans le Guide de l'utilisateur Amazon EC2 et le blog EC2-Classic Networking is Retiring – Here's How to Prepare (Se préparer au retrait de la mise en réseau EC2-Classic).

Pour connecter une instance de base de données Amazon RDS ne figurant pas dans un VPC à un serveur de réplication DMS et une instance de base de données dans un VPC, vous pouvez ClassicLink utiliser un serveur proxy.

ClassicLink vous permet de lier une instance de base de données EC2-Classic à un VPC de votre compte, dans la même région. AWS Une fois que vous avez créé le lien, l'instance DB source peut communiquer avec l'instance de réplication à l'intérieur du VPC à l'aide de leurs adresses IP privées.

Étant donné que l'instance de réplication du VPC ne peut pas accéder directement à l'instance de base de données source sur la plate-forme EC2-Classic à l'aide d' ClassicLinkun serveur proxy. Le serveur proxy connecte l'instance de base de données source au VPC contenant l'instance de réplication et l'instance de base de données cible. Le serveur proxy est utilisé ClassicLink pour se connecter au VPC. Le réacheminement de port sur le serveur proxy permet la communication entre l'instance de base de données source et l'instance de base de données cible dans le VPC.



Utilisation ClassicLink avec AWS Database Migration Service

Vous pouvez connecter une instance de base de données Amazon RDS qui n'est pas dans un VPC à un serveur de réplication DMS et AWS une instance de base de données qui se trouve dans un VPC. Pour ce faire, vous pouvez utiliser Amazon EC2 ClassicLink avec un serveur proxy.

La procédure suivante indique comment l'utiliser ClassicLink à cette fin. Cette procédure connecte une instance de base de données source Amazon RDS qui ne se trouve pas dans un VPC à un VPC contenant une instance de réplication DMS et AWS une instance de base de données cible.

- Créez une instance de réplication AWS DMS dans un VPC. (Toutes les instances de réplication sont créées dans des VPC.)
- Associez un groupe de sécurité de VPC à l'instance de réplication et à l'instance de base de données cible. Lorsque deux instances partagent un groupe de sécurité VPC, elles peuvent communiquer l'une avec l'autre par défaut.
- Configurez un serveur proxy sur une instance EC2 classique.
- Créez une connexion ClassicLink entre le serveur proxy et le VPC.
- Créez des points de terminaison AWS DMS pour les bases de données source et cible.
- Créez une tâche AWS DMS.

À utiliser pour ClassicLink migrer une base de données sur une instance de base de données ne faisant pas partie d'un VPC vers une base de données sur une instance de base de données d'un VPC

1. Créez une instance de réplication AWS DMS et attribuez un groupe de sécurité VPC :

   a. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.

      Si vous êtes connecté en tant qu'utilisateur AWS Identity and Access Management (IAM), assurez-vous de disposer des autorisations d'accès AWS DMS appropriées. Pour plus d'informations sur les autorisations requises pour la migration de base de données, consultez Autorisations IAM nécessaires pour utiliser AWS DMS.

b. Sur la page Tableau de bord, sélectionnez Instance de réplication. Suivez les instructions de la page Étape 1 : Créer une instance de réplication à l'aide de la console AWS DMS pour créer une instance de réplication.

c. Après avoir créé l'instance de réplication AWS DMS, ouvrez la console de service EC2. Choisissez Interfaces réseau dans le volet de navigation.

d. Choisissez le DMS NetworkInterface, puis choisissez Modifier les groupes de sécurité dans le menu Actions.

e. Choisissez le groupe de sécurité que vous voulez utiliser pour l'instance de réplication et l'instance de base de données cible.

2. Associez le groupe de sécurité de la dernière étape à l'instance de base de données cible.

a. Ouvrez la console de service Amazon RDS. Choisissez Instances dans le panneau de navigation.

b. Choisissez l'instance de base de données cible. Pour Actions d'instance, choisissez Modifier.

c. Pour le paramètre Groupe de sécurité, choisissez le groupe de sécurité que vous avez utilisé à l'étape précédente.

d. Choisissez Continuer, puis Modifier l'instance de base de données.

3. Étape 3 : Configurer un serveur proxy sur une instance EC2 classique à l'aide de NGINX. Utilisez une AMI de votre choix pour lancer une instance EC2 classique. L'exemple suivant est basée sur l'AMI Ubuntu Server 14.04 LTS (HVM).

   Pour configurer un serveur proxy sur une instance EC2 classique

   a. Connectez-vous à l'instance EC2 classique et installez NGINX à l'aide des commandes suivantes :

```
Prompt> sudo apt-get update
Prompt> sudo wget http://nginx.org/download/nginx-1.9.12.tar.gz
Prompt> sudo tar -xvzf nginx-1.9.12.tar.gz
Prompt> cd nginx-1.9.12
Prompt> sudo apt-get install build-essential
Prompt> sudo apt-get install libpcre3 libpcre3-dev
Prompt> sudo apt-get install zlib1g-dev
Prompt> sudo ./configure --with-stream
Prompt> sudo make
```

```
Prompt> sudo make install
```

b. Modifiez le fichier de démon NGINX `/etc/init/nginx.conf`, en utilisant le code suivant :

```
# /etc/init/nginx.conf – Upstart file

description "nginx http daemon"
author "email"

start on (filesystem and net-device-up IFACE=lo)
stop on runlevel [!2345]

env DAEMON=/usr/local/nginx/sbin/nginx
env PID=/usr/local/nginx/logs/nginx.pid

expect fork
respawn
respawn limit 10 5

pre-start script
        $DAEMON -t
        if [ $? -ne 0 ]
                then exit $?
        fi
end script

exec $DAEMON
```

c. Créez un fichier de configuration NGINX à l'adresse `/usr/local/nginx/conf/nginx.conf`. Dans le fichier de configuration, ajoutez les éléments suivants :

```
# /usr/local/nginx/conf/nginx.conf - NGINX configuration file

worker_processes  1;

events {
    worker_connections  1024;
}
```

```
stream {
  server {
    listen DB instance port number;
proxy_pass DB instance identifier:DB instance port number;
    }
}
```

d. À partir de la ligne de commande, démarrez NGINX à l'aide des commandes suivantes :

```
Prompt> sudo initctl reload-configuration
Prompt> sudo initctl list | grep nginx
Prompt> sudo initctl start nginx
```

4. Créez une ClassicLink connexion entre le serveur proxy et le VPC cible qui contient l'instance de base de données cible et l'instance de réplication :

   a. Ouvrez la console EC2 et choisissez l'instance EC2 Classic qui exécute le serveur proxy.

   b. Pour Actions, choisissez ClassicLink, puis choisissez Lier au VPC.

   c. Choisissez le groupe de sécurité que vous avez utilisé précédemment dans cette procédure.

   d. Choisissez Lien vers le VPC.

5. Étape 5 : Créez des points de terminaison AWS DMS à l'aide de la procédure décrite à l'adresse. Étape 2 : Spécifier les points de terminaison sources et cibles Veillez à utiliser le nom d'hôte DNS EC2 interne du proxy comme nom de serveur lorsque vous spécifiez le point de terminaison source.

6. Créez une tâche AWS DMS à l'aide de la procédure décrite dansÉtape 3 : Créer une tâche et migrer les données.

Créer groupe de sous-réseaux de réplication

Dans le cadre du réseau à utiliser pour la migration de bases de données, vous devez spécifier les sous-réseaux de votre cloud privé virtuel (VPC) que vous avez l'intention d'utiliser. Ce VPC doit être basé sur le service Amazon VPC. Un sous-réseau est une plage d'adresses IP dans votre VPC dans une Zone de disponibilité donnée. Ces sous-réseaux peuvent être répartis entre les zones de disponibilité de la AWS région où se trouve votre VPC.

Lorsque vous créez une instance de réplication ou un profil d'instance dans la console AWS DMS, vous pouvez utiliser le sous-réseau de votre choix.

Vous créez un groupe de sous-réseaux de réplication pour définir les sous-réseaux à utiliser. Vous devez spécifier les sous-réseaux dans au moins deux zones de disponibilité.

Pour créer un groupe de sous-réseaux de réplication

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.

   Si vous êtes connecté en tant qu'utilisateur IAM, vous devez détenir les autorisations appropriées pour accéder à AWS DMS. Pour plus d'informations sur les autorisations requises pour la migration de base de données, consultez Autorisations IAM nécessaires pour utiliser AWS DMS.
2. Dans le panneau de navigation, choisissez Subnet groups (Groupes de sous-réseaux).
3. Choisissez Créer groupe de sous-réseaux.
4. Sur la page Créer un groupe de sous-réseaux de réplication, spécifiez les informations de votre groupe de sous-réseaux de réplication. Le tableau suivant décrit les paramètres.

Option	Action
Nom	Entrez un nom pour le groupe de sous-réseaux de réplication, contenant entre 8 et 16 caractères ASCII imprimables (à l'exception de /, " et @). Le nom de votre compte doit être unique pour la AWS région que vous avez sélectionnée. Vous pouvez choisir d'ajouter une touche d'intelligence au nom, par exemple en incluant la AWS région et la tâche que vous effectuez **`DMS-default-VPC`** .
Description	Entrez une brève description du groupe de sous-réseaux de réplication.
VPC	Choisissez le VPC que vous voulez utiliser pour la migration des bases de données. Gardez à l'esprit que

Option	Action
	le VPC doit avoir au moins un sous-réseau dans au moins deux Zones de disponibilité.
Ajouter des sous-réseaux	Choisissez les sous-réseaux à inclure dans le groupe de sous-réseau de réplication. Vous devez choisir des sous-réseaux dans au moins deux zones de disponibilité.

5. Choisissez Créer groupe de sous-réseaux.

Résolution des points de terminaison de domaine à l'aide de DNS

Généralement, une instance de AWS DMS réplication utilise le résolveur DNS (Domain Name System) d'une instance Amazon EC2 pour résoudre les points de terminaison du domaine. Si vous avez besoin d'une résolution DNS, vous pouvez utiliser Amazon Route 53 Resolver. Pour plus d'informations sur l'utilisation du résolveur DNS Route 53, consultez Mise en route avec Route 53 Resolver.

Pour en savoir plus sur la façon d'utiliser votre propre serveur de noms sur site pour résoudre certains points de terminaison à l'aide d'Amazon Route 53 Resolver, consultez Utilisation de votre propre serveur de noms sur site.

Définition d'une clé de chiffrement pour une instance de réplication

AWS Le DMS chiffre le stockage utilisé par une instance de réplication et les informations de connexion du point de terminaison. Pour chiffrer le stockage utilisé par une instance de réplication, AWS DMS utilise un élément AWS KMS key propre à votre AWS compte. Vous pouvez afficher et gérer cette clé KMS avec AWS Key Management Service (AWS KMS). Vous pouvez utiliser la clé KMS par défaut de votre compte (`aws/dms`) ou une clé KMS que vous créez. Si vous possédez déjà une clé de AWS KMS chiffrement, vous pouvez également l'utiliser pour le chiffrement.

Vous pouvez spécifier votre propre clé de chiffrement en fournissant un identifiant de clé KMS pour chiffrer vos ressources AWS DMS. Lorsque vous spécifiez votre propre clé de chiffrement, le compte d'utilisateur utilisé pour effectuer la migration de base de données doit avoir accès à cette clé. Pour plus d'informations sur la création de vos propres clés de chiffrement et pour donner aux utilisateurs l'accès à une clé de chiffrement, consultez le Manuel du développeur AWS KMS.

Si vous ne spécifiez pas d'identifiant de clé KMS, AWS DMS utilise votre clé de chiffrement par défaut. KMS crée la clé de chiffrement par défaut pour AWS DMS pour votre AWS compte. Votre AWS compte possède une clé de chiffrement par défaut différente pour chaque AWS région.

Pour gérer les clés utilisées pour chiffrer vos ressources AWS DMS, vous utilisez. AWS KMS Vous pouvez le trouver AWS KMS en AWS Management Console recherchant KMS dans le volet de navigation.

AWS KMS combine du matériel et des logiciels sécurisés et hautement disponibles pour fournir un système de gestion des clés adapté au cloud. À l'aide de AWS KMS, vous pouvez créer des clés de chiffrement et définir les politiques qui contrôlent la manière dont ces clés peuvent être utilisées. AWS KMS prend en charge AWS CloudTrail, afin que vous puissiez auditer l'utilisation des clés pour vérifier que les clés sont utilisées de manière appropriée. Vos AWS KMS clés peuvent être utilisées en combinaison avec AWS DMS et d'autres AWS services pris en charge. Les services AWS pris en charge incluent Amazon RDS, Amazon S3, Amazon Elastic Block Store (Amazon EBS) et Amazon Redshift.

Lorsque vous avez créé vos ressources AWS DMS avec une clé de chiffrement spécifique, vous ne pouvez pas modifier la clé de chiffrement de ces ressources. Assurez-vous de déterminer vos exigences en matière de clé de chiffrement avant de créer vos ressources AWS DMS.

Création d'une instance de réplication

Dans le cadre de la migration d'une base de données, votre première tâche consiste à créer une instance de réplication. Cette instance de réplication nécessite suffisamment de stockage et de puissance de traitement pour effectuer les tâches que vous attribuez et pour migrer les données de la base de données source vers la base de données cible. La taille de cette instance varie en fonction de la quantité de données que vous devez migrer et des tâches que vous souhaitez que l'instance effectue. Pour plus d'informations sur les instances de réplication, consultez la page Utilisation d'une instance AWS DMS de réplication.

Pour créer une instance de réplication à l'aide de la AWS console

1. Choisissez les instances de réplication dans le volet de navigation de la AWS DMS console, puis sélectionnez Créer une instance de réplication.
2. Sur la page Créer une instance de réplication, spécifiez vos informations d'instance de réplication. Le tableau suivant décrit les paramètres que vous pouvez utiliser.

Option	Action
Nom	Entrez un nom pour l'instance de réplication contenant entre 8 et 16 caractères ASCII imprimables (à l'exception de /, " et @). Ce nom doit être unique pour votre compte dans la région AWS que vous avez sélectionnée. Vous pouvez choisir d'ajouter une touche d'intelligence au nom, par exemple en incluant la AWS région et la tâche que vous effectuez**`west2-mysql2mysql-instance1`** .
Amazon Resource Name (ARN) descriptif – Facultatif	Nom convivial pour remplacer l'ARN DMS par défaut. Vous ne pouvez pas le modifier après sa création.
Description	Entrez une brève description de l'instance de réplication.
Classe d'instance	Choisissez une classe d'instance avec la configuration dont vous avez besoin pour votre migration. Gardez à l'esprit que l'instance doit disposer de suffisamment de stockage, de réseau et de puissance de traitement pour mener à bien votre migration. Pour plus d'informations sur la façon de déterminer quelle classe d'instance est la mieux adaptée à votre migration, consultez la page Utilisation d'une instance AWS DMS de réplication.
Version du moteur	Dans la AWS DMS console, vous pouvez choisir la version de moteur prise en charge de votre choix. À partir du AWS CLI, l'instance de réplication exécute la dernière version non bêta du moteur de AWS DMS réplication, sauf si vous spécifiez une version du moteur différente dans le AWS CLI.

Option	Action
Haute disponibilité	Utilisez ce paramètre facultatif pour créer un réplica de secours de votre instance de réplication dans une autre Zone de disponibilité pour la prise en charge du basculement. Si vous avez l’intention d’utiliser la capture des données de modification (CDC) ou la réplication continue, vous devez activer cette option.

Option	Action
Stockage alloué (Gio)	Le stockage est principalement consommé par les fichiers journaux et les transactions mises en cache. Pour les transactions mises en cache, le stockage est utilisé uniquement lorsque les transactions mises en cache doivent être écrites sur disque. Par conséquen t, le AWS DMS n'utilise pas une quantité importante de stockage. Certaines exceptions sont les suivantes : • Les très grandes tables qui entraînent une charge de transaction importante. Le chargement d'une grande table peut prendre du temps. Les transacti ons mises en cache sont donc plus susceptib les d’être écrites sur le disque pendant un tel chargement. • Les tâches qui sont configurées pour marquer une pause avant de charger les transactions mises en cache. Dans ce cas, toutes les transactions sont mises en cache jusqu'à la fin du chargement complet pour toutes les tables. Avec cette configura tion, une quantité de stockage importante peut être utilisée par les transactions mises en cache. • Les tâches configurées avec des tables chargées dans Amazon Redshift. Toutefois, cette configura tion n’est pas un problème quand Amazon Aurora est la cible. Dans la plupart des cas, l'allocation de stockage par défaut est suffisante. Toutefois, il est toujours judicieux de prêter attention aux métriques liées au stockage. Veillez à augmenter votre capacité de stockage si vous constatez que vous consommez plus que ce qui vous est alloué par défaut.

Option	Action
Network type (Type de réseau)	DMS prend en charge le type de réseau de protocole d'adressage IPv4 et prend en charge les types de réseau de protocole d'adressage IPv4 et IPv6 en mode double pile. Si vous possédez des ressources qui doivent communiquer avec votre instance de réplication à l'aide d'un type de réseau de protocole d'adressage IPv6, utilisez le mode double pile. Pour en savoir plus sur les limitations en mode double pile, consultez Limitations pour les instances de base de données en réseau à double pile dans le Guide de l'utilisateur Amazon Relational Database Service.
VPC	Choisissez le VPC que vous souhaitez utiliser. Si votre base de données source ou cible se trouve dans un VPC, choisissez ce VPC. Si vos bases de données source et cible se trouvent dans des VPC différents, vérifiez qu'elles se trouvent toutes les deux dans des sous-réseaux publics et qu'elles sont accessibles publiquement. Choisissez ensuite le VPC dans lequel l'instance de réplication doit être située. L'instance de réplication doit être en métrique d'accéder aux données dans le VPC source. Si ni votre base de données source ni votre base de données cible ne se trouvent dans un VPC, choisissez un VPC dans lequel l'instance de réplication doit être située.

Option	Action
Groupe de sous-réseaux de réplication	Sélectionnez le groupe de sous-réseaux de réplication du VPC que vous avez sélectionnez, dans lequel vous souhaitez que l'instance de réplication soit créée. Si votre base de données source se trouve dans un VPC, sélectionnez le groupe de sous-réseaux qui contient la base de données source comme emplacement de votre instance de réplication. Pour plus d'informations sur les groupes de sous-réseaux de réplication, consultez la page Créer groupe de sous-réseaux de réplication.
Accessible publiquement	Choisissez cette option si vous souhaitez que l'instance de réplication soit accessible à partir d'Internet. La valeur par défaut est accessible publiquement, et une fois l'option choisie, vous ne pouvez pas la modifier après avoir créé l'instance de réplication.

3. Cliquez sur l'onglet Avancé pour définir des valeurs pour les paramètres de réseau et de chiffrement si vous en avez besoin. Le tableau suivant décrit les paramètres.

Option	Action
Zone de disponibilité	Sélectionnez la Zone de disponibilité dans laquelle se trouve votre base de données source.
Groupes de sécurité VPC	L'instance de réplication est créée dans un VPC. Si la base de données source se trouve dans un VPC, choisissez le groupe de sécurité de VPC qui permet d'accéder à l'instance de base de données où réside la base de données.
Clé KMS	Sélectionnez la clé de chiffrement à utiliser pour chiffrer les informations de connexion et de stockage de réplication. Si vous choisissez (par défaut) aws/dms, la clé par défaut AWS Key Management Service (AWS KMS) associée à votre compte et à votre AWS

Option	Action
	région est utilisée. Une description et votre numéro de compte sont affichés, ainsi que l'ARN de la clé. Pour plus d'informations sur l'utilisation de la clé chiffrement, consultez la page Définition d'une clé de chiffrement et spécification des autorisations AWS KMS.

4. Spécifiez les paramètres de Maintenance. Le tableau suivant décrit les paramètres. Pour plus d'informations sur les paramètres de maintenance, consultez Utilisation de la fenêtre de maintenance AWS DMS.

Option	Action
Mise à niveau automatique de la version	AWS DMS ne fait pas la différence entre les versions majeures et mineures. Par exemple, la mise à niveau de la version 3.4.x vers 3.5.x n'est pas considéré e comme une mise à niveau majeure. Toutes les modifications doivent donc être rétrocompatibles. Quand l'option Mise à niveau automatique de la version est activée, DMS met automatiquement à niveau la version de l'instance de réplication pendant la fenêtre de maintenance si elle est obsolète. Lorsque cette option AutoMinorVersionUpgradeest activée, DMS utilise la version par défaut actuelle du moteur lorsque vous créez une instance de réplicati on. Par exemple, si vous définissez Version du moteur sur un numéro de version inférieur à celui de la version par défaut actuelle, DMS utilise la version par défaut. S'il AutoMinorVersionUpgraden'est pas activé lorsque vous créez une instance de réplication, DMS utilise la version du moteur spécifiée par le paramètre de version du moteur.

Option	Action
Fenêtre de maintenance	Choisissez un intervalle de temps hebdomadaire, au format Universal Coordinated Time (UTC), pendant lequel a lieu la maintenance du système. Par défaut : une fenêtre de 30 minutes sélectionnée au hasard parmi une plage de 8 heures par AWS région, survenant un jour aléatoire de la semaine.

5. Choisissez Créer instance de réplication.

Modification d'une instance de réplication

Vous pouvez modifier les paramètres d'une instance de réplication pour, par exemple, modifier la classe d'instance ou augmenter l'espace de stockage.

Lorsque vous modifiez une instance de réplication, vous pouvez appliquer les modifications immédiatement. Pour appliquer immédiatement les modifications, choisissez l'option Appliquer immédiatement les modifications dans la AWS Management Console. Vous pouvez également utiliser le `--apply-immediately` paramètre lorsque vous appelez AWS CLI le ou définissez le `ApplyImmediately` paramètre sur `true` lorsque vous utilisez l'API DMS.

Si vous ne choisissez pas d'appliquer les modifications immédiatement, les modifications sont placées dans la file d'attente des modifications en attente. Au cours de la fenêtre de maintenance suivante, les modifications en attente sont appliquées.

> **Note**
>
> Si vous choisissez d'appliquer les modifications immédiatement, les modifications placées dans la file d'attente des modifications en attente sont également appliquées. Si des modifications en attente ont besoin d'un temps d'arrêt, le choix de l'option Appliquer immédiatement les modifications peut entraîner un temps d'arrêt imprévu.

Pour modifier une instance de réplication à l'aide de la AWS console

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.

2. Dans le volet de navigation, sélectionnez Instances de réplication.
3. Choisissez l'instance de réplication que vous souhaitez modifier. Le tableau suivant décrit les modifications que vous pouvez effectuer.

Option	Action
Nom	Vous pouvez modifier le nom de l'instance de réplication. Entrez un nom pour l'instance de réplication contenant entre 8 et 16 caractères ASCII imprimables (à l'exception de /, " et @). Ce nom doit être unique pour votre compte dans la région AWS que vous avez sélectionnée. Vous pouvez choisir d'ajouter une touche d'intelligence au nom, par exemple en incluant la AWS région et la tâche que vous effectuez**`west2-mysql2mysql-instance1`** .
Description	Révisez ou entrez une brève description de l'instance de réplication.
Classe d'instance	Vous pouvez modifier la classe d'instance. Choisissez une classe d'instance avec la configuration dont vous avez besoin pour votre migration. La modification de la classe d'instance entraîne le redémarrage de l'instance de réplication. Ce redémarrage a lieu au cours de la fenêtre de maintenance suivante ou peut se produire immédiatement si vous choisissez l'option Appliquer les modifications immédiatement. Pour plus d'informations sur la façon de déterminer quelle classe d'instance est la mieux adaptée à votre migration, consultez la page Utilisation d'une instance AWS DMS de réplication.
Version du moteur	Vous pouvez mettre à niveau la version du moteur utilisée par l'instance de réplication. La mise à niveau de la version du moteur de réplication entraîne l'arrêt de l'instance de réplication pendant la mise à niveau.

Option	Action
Multi-AZ	Vous pouvez modifier cette option pour créer un réplica de secours de votre instance de réplication dans une autre Zone de disponibilité pour la prise en charge du basculement ou supprimer cette option. Si vous avez l'intention d'utiliser la capture de données modifiées (CDC), la réplication continue, vous devez activer cette option.

Option	Action
Stockage alloué (Gio)	Le stockage est principalement consommé par les fichiers journaux et les transactions mises en cache. Pour les transactions mises en cache, le stockage est utilisé uniquement lorsque les transactions mises en cache doivent être écrites sur disque. Par conséquen t, le AWS DMS n'utilise pas une quantité importante de stockage. Certaines exceptions sont les suivantes : • Les très grandes tables qui entraînent une charge de transaction importante. Le chargement d'une grande table peut prendre du temps. Les transacti ons mises en cache sont donc plus susceptib les d’être écrites sur le disque pendant un tel chargement. • Les tâches qui sont configurées pour marquer une pause avant de charger les transactions mises en cache. Dans ce cas, toutes les transactions sont mises en cache jusqu'à la fin du chargement complet pour toutes les tables. Avec cette configura tion, une quantité de stockage importante peut être utilisée par les transactions mises en cache. • Les tâches configurées avec des tables chargées dans Amazon Redshift. Toutefois, cette configura tion n’est pas un problème quand Amazon Aurora est la cible. Dans la plupart des cas, l'allocation de stockage par défaut est suffisante. Cependant, il est toujours préférable de faire attention aux métriques associées au stockage et de dimensionner votre stockage si vous constatez que vous consommez plus que l'allocation par défaut.

Option	Action
Network type (Type de réseau)	DMS prend en charge le type de réseau de protocole d'adressage IPv4 et prend en charge les types de réseau de protocole d'adressage IPv4 et IPv6 en mode double pile. Si vous possédez des ressource s qui doivent communiquer avec votre instance de réplication à l'aide d'un type de réseau de protocole d'adressage IPv6, choisissez le mode Double pile. Pour en savoir plus sur les limitations en mode double pile, consultez Limitations pour les instances de base de données en réseau à double pile dans le Guide de l'utilisateur Amazon Relational Database Service.
Groupes de sécurité VPC	L'instance de réplication est créée dans un VPC. Si la base de données source se trouve dans un VPC, choisissez le groupe de sécurité de VPC qui permet d'accéder à l'instance de base de données où réside la base de données.

Option	Action
Mise à niveau automatique de la version	AWS DMS ne fait pas la différence entre les versions majeures et mineures. Par exemple, la mise à niveau de la version 3.4.x vers 3.5.x n'est pas considéré e comme une mise à niveau majeure. Toutes les modifications doivent donc être rétrocompatibles. Quand l'option Mise à niveau automatique de la version est activée, DMS met automatiquement à niveau la version de l'instance de réplication pendant la fenêtre de maintenance si elle est obsolète. Quand l'option Mise à niveau automatique de la version est activée, DMS utilise la version actuelle du moteur par défaut lorsque vous créez une instance de réplication. Par exemple, si vous définissez Version du moteur sur un numéro de version inférieur à celui de la version par défaut actuelle, DMS utilise la version par défaut. Si l'option Mise à niveau automatique de la version n'est pas activée lorsque vous créez une instance de réplication, DMS utilise la version du moteur spécifiée par le paramètre Version du moteur.
Fenêtre de maintenance	Choisissez un intervalle de temps hebdomadaire, au format Universal Coordinated Time (UTC), pendant lequel a lieu la maintenance du système. Par défaut : une fenêtre de 30 minutes sélectionnée au hasard parmi une plage de 8 heures par AWS région, survenant un jour aléatoire de la semaine.

Option	Action
Appliquer immédiatement les modifications	Choisissez cette option pour appliquer immédiate ment les modifications que vous avez apportées. En fonction des paramètres que vous choisissez, la sélection de cette option peut entraîner un redémarra ge immédiat de l'instance de réplication. Si vous choisissez Tester la connexion alors qu’ AWS DMS applique des modifications, un message d’erreur s’affiche. Après avoir AWS DMS appliqué les modificat ions à votre instance de réplication, choisissez à nouveau Tester la connexion.
Appliquer les modifications au cours de la prochaine fenêtre de maintenance planifiée	Choisissez cette option si vous souhaitez que DMS attende la prochaine fenêtre de maintenance planifiée pour appliquer vos modifications.

redémarrage d'une instance de réplication.

Vous pouvez redémarrer une instance AWS DMS de réplication pour redémarrer le moteur de réplication. Le redémarrage entraîne une interruption momentanée de l'instance de réplication. À cette occasion, le statut de l'instance est défini sur Redémarrage. Si l' AWS DMS instance est configurée pour le mode multi-AZ, le redémarrage peut être effectué avec un basculement. Un AWS DMS événement est créé lorsque le redémarrage est terminé.

Si votre AWS DMS instance est un déploiement multi-AZ, vous pouvez forcer un basculement planifié d'une zone de AWS disponibilité à une autre lors du redémarrage. Lorsque vous forcez un basculement planifié de votre AWS DMS instance, AWS DMS ferme les connexions actives sur l'instance actuelle avant de passer automatiquement à une instance de secours dans une autre zone de disponibilité. Le redémarrage à l'aide d'un basculement planifié vous permet de simuler un événement de basculement planifié d'une AWS DMS instance, par exemple lors du dimensionnement de la classe d'instance de réplication.

> **Note**
>
> Lorsqu'un redémarrage force un basculement d'une zone de disponibilité à une autre, le changement de zone de disponibilité peut ne pas être reflété pendant quelques minutes. Ce décalage apparaît dans et dans les appels à l' AWS DMS API AWS CLI and. AWS Management Console

Si des tâches de migration s'exécutent sur l'instance de réplication au moment du redémarrage, aucune perte de données ne se produit, mais la tâche s'arrête et son statut passe à l'état d'erreur.

Si les tables de la tâche de migration sont en cours de chargement en bloc (phase de chargement complet) et n'ont pas encore démarré, elles passent à un état d'erreur. En revanche, les tables qui sont complètes à ce moment-là restent à un état complet. Lorsqu'un redémarrage se produit pendant la phase de chargement complet, nous vous recommandons d'effectuer l'une des étapes ci-dessous.

- Supprimer de la tâche les tables dont l'état est complet, puis redémarrer la tâche avec les tables restantes.
- Créer une nouvelle tâche avec des tables en état d'erreur et des tables en attente.

Si les tables comprises dans la tâche de migration se trouvent dans la phase de réplication continue, la tâche reprend une fois que le redémarrage est terminé.

Vous ne pouvez pas redémarrer votre instance de AWS DMS réplication si son statut n'est pas disponible. Votre AWS DMS instance peut être indisponible pour plusieurs raisons, telles qu'une modification demandée précédemment ou une action liée à la fenêtre de maintenance. Le temps nécessaire au redémarrage d'une instance de AWS DMS réplication est généralement court (moins de 5 minutes).

Redémarrage d'une instance de réplication à l'aide de la console AWS

Pour redémarrer une instance de réplication, utilisez la AWS console.

Pour redémarrer une instance de réplication à l'aide de la AWS console

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Dans le volet de navigation, sélectionnez Instances de réplication.

3. Choisissez l'instance de réplication que vous souhaitez redémarrer.
4. Choisissez Redémarrer. La boîte de dialogue Redémarrer l'instance de réplication s'ouvre.
5. Cochez la case Redémarrer avec le basculement planifié ? si vous avez configuré votre instance de réplication pour le déploiement multi-AZ et que vous souhaitez basculer vers une autre zone de disponibilité AWS .
6. Choisissez Redémarrer.

Redémarrage d'une instance de réplication via l'interface de ligne de commande

Pour redémarrer une instance de réplication, utilisez la AWS CLI `reboot-replication-instance`commande avec le paramètre suivant :

- `--replication-instance-arn`

Example Exemple de redémarrage simple

L' AWS CLI exemple suivant redémarre une instance de réplication.

```
aws dms reboot-replication-instance \
--replication-instance-arn arn of my rep instance
```

Example Exemple de redémarrage simple avec basculement

L' AWS CLI exemple suivant redémarre une instance de réplication avec basculement.

```
aws dms reboot-replication-instance \
--replication-instance-arn arn of my rep instance \
--force-planned-failover
```

Redémarrage d'une instance de réplication via l'API

Pour redémarrer une instance de réplication, utilisez l'`RebootReplicationInstance`action AWS DMS API avec les paramètres suivants :

- `ReplicationInstanceArn = arn of my rep instance`

Example Exemple de redémarrage simple

L'exemple de code suivant redémarre une instance de réplication.

```
https://dms.us-west-2.amazonaws.com/
?Action=RebootReplicationInstance
&DBInstanceArn=arn of my rep instance
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-09-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140425/us-east-1/dms/aws4_request
&X-Amz-Date=20140425T192732Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=1dc9dd716f4855e9bdf188c70f1cf9f6251b070b68b81103b59ec70c3e7854b3
```

Example Exemple de redémarrage simple avec basculement

L'exemple de code suivant redémarre une instance de réplication et bascule vers une autre zone de AWS disponibilité.

```
https://dms.us-west-2.amazonaws.com/
?Action=RebootReplicationInstance
&DBInstanceArn=arn of my rep instance
&ForcePlannedFailover=true
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-09-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140425/us-east-1/dms/aws4_request
&X-Amz-Date=20140425T192732Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=1dc9dd716f4855e9bdf188c70f1cf9f6251b070b68b81103b59ec70c3e7854b3
```

Supprimez une instance de réplication.

Vous pouvez supprimer une instance de AWS DMS réplication lorsque vous avez fini de l'utiliser. Si vous avez des tâches de migration qui utilisent l'instance de réplication, vous devez arrêter et supprimer ces tâches avant de supprimer l'instance de réplication.

Si vous fermez votre AWS compte, toutes les AWS DMS ressources et configurations associées à celui-ci sont supprimées au bout de deux jours. Ces ressources incluent toutes les instances de

réplication, la configuration des points de terminaison source et cible, les tâches de réplication et les certificats SSL. Si, au bout de deux jours, vous décidez de AWS DMS réutiliser, vous recréez les ressources dont vous avez besoin.

Si votre instance de réplication répond à tous les critères de suppression et qu'elle conserve son statut `DELETING` pendant une période prolongée, contactez le support pour résoudre le problème.

Suppression d'une instance de réplication à l'aide de la AWS console

Pour supprimer une instance de réplication, utilisez la AWS console.

Pour supprimer une instance de réplication à l'aide de la AWS console

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Dans le volet de navigation, sélectionnez Instances de réplication.
3. Choisissez l'instance de réplication que vous souhaitez supprimer.
4. Sélectionnez Delete (Supprimer).
5. Dans la boîte de dialogue , , choisissez Supprimer.

Suppression d'une instance de réplication à l'aide de l'interface de ligne de commande

Pour supprimer une instance de réplication, utilisez la AWS CLI `delete-replication-instance`commande avec le paramètre suivant :

- `--replication-instance-arn`

Example Exemple de suppression

L' AWS CLI exemple suivant supprime une instance de réplication.

```
aws dms delete-replication-instance \
--replication-instance-arn arn of my rep instance
```

Suppression d'une instance de réplication à l'aide de l'API

Pour supprimer une instance de réplication, utilisez l'`DeleteReplicationInstance`action AWS DMS API avec les paramètres suivants :

- `ReplicationInstanceArn = arn of my rep instance`

Example Exemple de suppression

L'exemple de code suivant supprime une instance de réplication.

```
https://dms.us-west-2.amazonaws.com/
?Action=DeleteReplicationInstance
&DBInstanceArn=arn of my rep instance
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-09-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140425/us-east-1/dms/aws4_request
&X-Amz-Date=20140425T192732Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=1dc9dd716f4855e9bdf188c70f1cf9f6251b070b68b81103b59ec70c3e7854b3
```

Utilisation de la fenêtre de maintenance AWS DMS

Chaque instance de AWS DMS réplication dispose d'une fenêtre de maintenance hebdomadaire au cours de laquelle toutes les modifications système disponibles sont appliquées. Vous pouvez considérer le créneau de maintenance comme une occasion de contrôler le moment où les modifications et les correctifs logiciels sont appliqués.

S'il est AWS DMS déterminé qu'une maintenance est requise au cours d'une semaine donnée, elle a lieu pendant la période de maintenance de 30 minutes que vous avez choisie lors de la création de l'instance de réplication. AWS DMS effectue la plupart des opérations de maintenance pendant la fenêtre de maintenance de 30 minutes. Toutefois, une plus longue durée peut être nécessaire pour de plus grandes modifications.

Effet de la maintenance sur les tâches de migration existantes

Lorsqu'une tâche de AWS DMS migration est exécutée sur une instance, les événements suivants se produisent lorsqu'un correctif est appliqué :

- Si les tables dans la tâche de migration se trouvent dans la phase de réplication des modifications en continu (CDC), AWS DMS arrête la tâche pendant un moment, puis la reprend après l'application du correctif. La migration reprend ensuite à partir du point où elle a été interrompue lorsque le correctif a été appliqué.

- S'il s' AWS DMS agit de migrer une table dans le cadre d'une tâche de migration de données existantes ou de migration de données existantes et de réplication des modifications en cours, DMS arrête puis redémarre la migration pour toutes les tables en phase de chargement complet pendant l'application du correctif. DMS arrête et reprend également toutes les tables en phase CDC pendant l'application du correctif.

Modification de la configuration de la fenêtre de maintenance

Vous pouvez modifier la période de maintenance à l'aide de l'API AWS Management Console, de ou de l' AWS DMS API. AWS CLI

Modification de la configuration de la fenêtre de maintenance à l'aide de la console

Vous pouvez modifier l'horaire de la fenêtre de maintenance à l'aide de l' AWS Management Console.

Pour modifier la fenêtre de maintenance préférée à l'aide de la console

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Dans le volet de navigation, sélectionnez Instances de réplication.
3. Choisissez l'instance de réplication que vous souhaitez modifier, puis sélectionnez Modifier.
4. Développez la section Maintenance et choisissez la date et l'heure de votre fenêtre de maintenance.
5. Choisissez Apply changes immediately.
6. Sélectionnez Modifier.

Modification de la configuration de la fenêtre de maintenance à l'aide de l'interface de ligne de commande

Pour ajuster la fenêtre de maintenance préférée, utilisez la AWS CLI `modify-replication-instance`commande avec les paramètres suivants.

- `--replication-instance-identifier`
- `--preferred-maintenance-window`

Example

L' AWS CLI exemple suivant définit la fenêtre de maintenance sur les mardis, de 4 h 00 à 4 h 30. UTC.

```
aws dms modify-replication-instance \
--replication-instance-identifier myrepinstance \
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

Modification de la configuration de la fenêtre de maintenance à l'aide de l'API

Pour ajuster la fenêtre de maintenance préférée, utilisez l'ModifyReplicationInstanceaction AWS DMS API avec les paramètres suivants.

- `ReplicationInstanceIdentifier = myrepinstance`
- `PreferredMaintenanceWindow = Tue:04:00-Tue:04:30`

Example

L'exemple de code suivant définit la fenêtre de maintenance pour qu'elle ait lieu tous les mardis entre 04 h 00 et 04 h 30. UTC.

```
https://dms.us-west-2.amazonaws.com/
?Action=ModifyReplicationInstance
&DBInstanceIdentifier=myrepinstance
&PreferredMaintenanceWindow=Tue:04:00-Tue:04:30
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-09-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140425/us-east-1/dms/aws4_request
&X-Amz-Date=20140425T192732Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=1dc9dd716f4855e9bdf188c70f1cf9f6251b070b68b81103b59ec70c3e7854b3
```

Utilisation des points de terminaison AWS DMS

Un point de terminaison fournit la connexion, le type de magasin de données et les informations d'emplacement de votre magasin de données. AWS Database Migration Service utilise ces informations pour se connecter à un magasin de données et migrer les données depuis un point de terminaison source vers un point de terminaison cible. Vous pouvez spécifier des attributs de connexion supplémentaires pour un point de terminaison à l'aide des paramètres de point de terminaison. Ces paramètres peuvent contrôler la journalisation, la taille de fichier et d'autres paramètres ; pour plus d'informations sur les paramètres de point de terminaison, consultez la section de la documentation relative à votre magasin de données.

Ci-dessous, vous trouverez plus d'informations sur les points de terminaison.

Rubriques

- Création de points de terminaison source et cible
- Sources pour la migration des données
- Cibles pour la migration des données
- Configuration de points de terminaison de VPC en tant que points de terminaison sources et cibles AWS DMS
- Instructions DDL prises en charge par AWS DMS

Création de points de terminaison source et cible

Vous pouvez créer des points de terminaison source et cible lorsque vous créez votre instance de réplication ou vous pouvez créer les points de terminaison après que votre instance de réplication a été créée. Les magasins de données source et cible peuvent se trouver sur une instance Amazon Elastic Compute Cloud (Amazon EC2), une instance de base de données Amazon Relational Database Service (Amazon RDS) ou une base de données sur site. (Notez que l'un de vos points de terminaison doit se trouver sur un service AWS. Vous ne pouvez pas utiliser AWS DMS pour migrer depuis une base de données sur site vers une autre base de données sur site.)

La procédure suivante suppose que vous avez choisi l'assistant de la console AWS DMS. Notez que vous pouvez également effectuer cette étape en sélectionnant Points de terminaison dans le volet de navigation de la console AWS DMS, puis en sélectionnant Créer un point de terminaison. Lorsque vous utilisez l'assistant de la console, vous créez les points de terminaison source et cible

sur la même page. Lorsque vous n'utilisez pas l'assistant de la console, vous créez chaque point de terminaison séparément.

Pour spécifier un point de terminaison de base de données source ou cible à l'aide de la console AWS

1. Sur la page Connecter les points de terminaison des bases de données source et cible, spécifiez vos informations de connexion pour la base de données source ou cible. Le tableau suivant décrit les paramètres.

Pour cette option	Faire ceci
Type de point de terminaison	Choisissez si ce point de terminaison est le point de terminaison source ou cible.
Select RDS DB Instance (Sélectionner une instance de base de données RDS)	Choisissez cette option si le point de terminaison est une instance de base de données Amazon RDS.
Identificateur de point de terminaison	Tapez le nom que vous voulez utiliser pour identifier le point de terminaison. Vous pouvez inclure dans le nom du type de point de terminaison, par exemple **`oracle-source`** ou **`PostgreSQL-target`** . Le nom doit être unique pour toutes les instances de réplication.
Moteur source et Moteur cible	Choisissez le type de moteur de base de données qui correspond au point de terminaison.

Pour cette option	Faire ceci
Accès à la base de données du point de terminaison	Choisissez l'option que vous voulez utiliser pour spécifier les informations d'identification de la base de données du point de terminaison : • Choisissez AWS Secrets Manager : utilisez les secrets définis dans AWS Secrets Manager pour fournir secrètement vos informations d'identification, comme indiqué ci-dessous. Pour plus d'informations sur la création de ces secrets et sur les rôles d'accès secrets qui permettent à AWS DMS d'y accéder, consultez Utilisation de secrets pour accéder aux points de terminaison AWS Database Migration Service. • Renseignez les informations d'accès manuellement : utilisez des informations d'identification en texte clair que vous entrez directement, comme indiqué ci-dessous.
Choisissez AWS Secrets Manager	Définissez les informations d'identification secrètes suivantes.
ID secret	Saisissez l'Amazon Resource Name (ARN) complet, l'ARN partiel ou le nom convivial d'un secret que vous avez créé dans AWS Secrets Manager pour accéder à la base de données du point de terminaison.
Rôle IAM	Tapez l'ARN d'un rôle d'accès secret que vous avez créé dans IAM pour permettre à AWS DMS d'accéder en votre nom au secret identifié par ID secret. Pour en savoir plus sur la création d'un rôle d'accès secret, consultez Utilisation de secrets pour accéder aux points de terminaison AWS Database Migration Service.

Pour cette option	Faire ceci
ID secret pour Oracle Automatic Storage Management (ASM)	(Pour les points de terminaison sources Oracle utilisant Oracle ASM uniquement) Saisissez l'Amazon Resource Name (ARN) complet, l'ARN partiel ou le nom convivial d'un secret que vous avez créé dans AWS Secrets Manager pour l'accès à Oracle ASM. Ce secret est généralement créé pour accéder à Oracle ASM sur le même serveur que le secret identifié par ID secret.
Rôle IAM pour Oracle ASM	(Pour les points de terminaison sources Oracle utilisant Oracle ASM uniquement) Saisissez l'ARN d'un rôle d'accès secret que vous avez créé dans IAM pour fournir à AWS DMS l'accès en votre nom au secret identifié par ID secret pour Oracle Automatic Storage Management (ASM).
Renseignez les informations d'accès manuellement	Définissez les informations d'identification en texte clair suivantes.
Server name	Tapez le nom du serveur. Pour une base de données sur site, cela peut être l'adresse IP ou le nom d'hôte public. Pour une instance de base de données Amazon RDS, cela peut être le point de terminaison (également appelé nom DNS) pour l'instance de base de données, comme **`mysqlsrvinst.abcd12345678.us-west-2.rds.amazonaws.com`** .
Port	Saisissez le port utilisé par la base de données.
Mode SSL (Secure Socket Layer)	Sélectionnez un mode SSL si vous souhaitez activer le chiffrement de connexion pour ce point de terminaison. Selon le mode que vous sélectionnez, vous pouvez être invité à fournir les informations concernant le certificat et le certificat du serveur.

Pour cette option	Faire ceci
Nom utilisateur	Tapez le nom d'utilisateur avec les autorisations requises pour autoriser la migration des données. Pour plus d'informations sur les autorisations requises, consultez la section relative à la sécurité pour le moteur de base de données source ou cible dans ce guide de l'utilisateur.
Mot de passe :	Tapez le mot de passe du compte avec les autorisat ions requises. Les mots de passe des points de terminaison AWS DMS sources et cibles sont soumis à des restrictions de caractères, en fonction du moteur de base de données. Pour plus d'informations, consultez le tableau suivant.
Nom de base de données	Pour certains moteurs de base de données, le nom de la base de données que vous souhaitez utiliser comme base de données des points de terminaison.

Le tableau suivant répertorie les caractères non pris en charge dans les mots de passe des terminaux et les secrets du gestionnaire de secrets pour les moteurs de base de données répertoriés. Si vous souhaitez utiliser des virgules (,) dans vos mots de passe de point de terminaison, utilisez le support Secrets Manager fourni dans AWS DMS pour authentifier l'accès à vos instances AWS DMS. Pour plus d'informations, consultez Utilisation de secrets pour accéder aux points de terminaison AWS Database Migration Service.

Pour ce moteur de base de données	Les caractères suivants ne sont pas pris en charge dans le mot de passe d'un terminal et dans les secrets du gestionnaire de secrets
Tous	`{ }`
Microsoft Azure, en tant que source uniquement	`;`

Pour ce moteur de base de données	Les caractères suivants ne sont pas pris en charge dans le mot de passe d'un terminal et dans les secrets du gestionnaire de secrets
Microsoft SQL Server	, ;
Compatible MySQL, y compris MySQL, MariaDB et Amazon Aurora MySQL	;
Oracle	,
PostgreSQL, Amazon Aurora PostgreSQL-Compatible Edition et Amazon Aurora sans serveur en tant que cible uniquement pour Aurora PostgreSQL-Compatible Edition	; + %
Amazon Redshift, en tant que cible uniquement	, ;

2. Choisissez Paramètres de point de terminaison et AWS KMS key si vous en avez besoin. Vous pouvez tester la connexion au point de terminaison en sélectionnant Exécuter test. Le tableau suivant décrit les paramètres.

Pour cette option	Faire ceci
Paramètres de point de terminaison	Sélectionnez des paramètres de connexion supplémentaires ici. Pour plus d'informations sur les paramètres de point de terminaison, consultez la section de la documentation relative à votre moteur source ou à votre moteur cible (spécifiée à l'étape 1). Pour un point de terminaison source Oracle qui utilise Oracle ASM, si vous choisissez Renseignez les informations d'accès manuellement à l'étape 1, vous devrez peut-être également saisir le paramètre

Pour cette option	Faire ceci
	de point de terminaison pour spécifier les informati ons d'identification de l'utilisateur Oracle ASM. Pour plus d'informations sur ces paramètres de point de terminaison Oracle ASM, consultez Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC.
AWS KMS key	Sélectionnez la clé de chiffrement à utiliser pour chiffrer les informations de connexion et de stockage de réplication. Si vous choisissez (Par défaut) aws/ dms, la clé AWS Key Management Service (AWS KMS) par défaut associée à votre compte et votre région AWS est utilisée. Pour plus d'informations sur l'utilisation de la clé chiffrement, consultez la page Définition d'une clé de chiffrement et spécification des autorisations AWS KMS.
Test endpoint connection (Tester la connexion du point de terminaison) (facultatif)	Ajoutez le VPC et le nom de l'instance de réplication. Pour tester la connexion, choisissez Exécuter test.

Sources pour la migration des données

AWS Database Migration Service (AWS DMS) peut utiliser un grand nombre des moteurs de données les plus populaires comme source pour la réplication des données. La source de base de données peut être un moteur autogéré s'exécutant sur une instance Amazon EC2 ou une base de données sur site. Ou, ce peut être une source de données sur un service AWS tel qu'Amazon RDS ou Amazon S3.

Pour une liste complète des sources valides, consultez Sources pour AWS DMS.

Rubriques

- Utilisation d'une base de données Oracle comme source pour AWS DMS
- Utilisation d'une base de données Microsoft SQL Server comme source pour AWS DMS
- Utilisation de Microsoft Azure SQL Database comme source pour AWS DMS

- Utilisation de Microsoft Azure SQL Managed Instance en tant que source pour AWS DMS
- Utilisation du serveur flexible Microsoft Azure Database pour PostgreSQL en tant que source pour AWS DMS
- Utilisation du serveur flexible Microsoft Azure Database pour MySQL en tant que source pour AWS DMS
- Utilisation d'OCI MySQL Heatwave en tant que source pour AWS DMS
- Utilisation de Google Cloud pour MySQL en tant que source pour AWS DMS
- Utilisation de Google Cloud pour PostgreSQL en tant que source pour AWS DMS
- Utilisation d'une base de données PostgreSQL en tant que source AWS DMS
- Utilisation d'une base de données compatible MySQL comme source pour AWS DMS
- Utilisation d'une base de données SAP ASE comme source pour AWS DMS
- Utilisation de MongoDB comme source pour AWS DMS
- Utilisation d'Amazon DocumentDB (compatible avec MongoDB) en tant que source pour AWS DMS
- Utilisation d'Amazon S3 en tant que source pour AWS DMS
- Utilisation d'IBM Db2 pour Linux, Unix, Windows et de la base de données Amazon RDS (Db2 LUW) comme source pour AWS DMS
- Utilisation de bases de données IBM Db2 for z/OS en tant que source pour AWS DMS

Utilisation d'une base de données Oracle comme source pour AWS DMS

Vous pouvez migrer les données d'une ou de plusieurs bases de données Oracle à l'aide de AWS DMS. Avec une base de données Oracle en tant que source, vous pouvez migrer des données vers l'une des cibles prises en charge par AWS DMS.

AWS DMS prend en charge les éditions de base de données Oracle suivantes :

- Oracle Enterprise Edition
- Oracle Standard Edition
- Édition Oracle Express
- Édition Oracle Personal

Pour plus d'informations sur les versions des bases de données Oracle AWS DMS prises en charge en tant que source, consultezSources pour AWS DMS.

Vous pouvez utiliser SSL (Secure Sockets Layer) pour chiffrer les connexions entre votre point de terminaison Oracle et votre instance de réplication. Pour plus d'informations sur l'utilisation de SSL avec un point de terminaison Oracle, consultez Prise en charge de SSL pour un point de terminaison Oracle.

AWS DMS prend en charge l'utilisation du chiffrement transparent des données (TDE) Oracle pour chiffrer les données au repos dans la base de données source. Pour plus d'informations sur l'utilisation de TDE Oracle avec un point de terminaison source Oracle, consultez Méthodes de chiffrement prises en charge pour l'utilisation d'Oracle comme source pour AWS DMS.

AWS prend en charge l'utilisation de TLS version 1.2 et ultérieure avec les points de terminaison Oracle (et tous les autres types de points de terminaison) et recommande d'utiliser TLS version 1.3 ou ultérieure.

Procédez comme suit pour configurer une base de données Oracle en tant que point de terminaison AWS DMS source :

1. Créez un utilisateur Oracle doté des autorisations appropriées pour accéder AWS DMS à votre base de données source Oracle.
2. Créez un point de terminaison source Oracle conforme à la configuration de base de données Oracle que vous avez choisie. Pour créer une full-load-only tâche, aucune autre configuration n'est nécessaire.
3. Pour créer une tâche qui gère la capture des données de modification (une tâche uniquement CDC ou une tâche à chargement complet et une tâche CDC), choisissez Oracle LogMiner ou AWS DMS Binary Reader pour capturer les modifications de données. Le choix LogMiner de Binary Reader détermine certaines des autorisations et options de configuration ultérieures. Pour une comparaison entre Binary Reader LogMiner et Binary Reader, consultez la section suivante.

> **Note**
>
> Pour plus d'informations sur les tâches de chargement complet, les tâches de CDC uniquement et les tâches de chargement complet et de CDC, consultez Création d'une tâche

Pour plus de détails sur l'utilisation des bases de données source Oracle AWS DMS, consultez les sections suivantes.

Rubriques

- Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC
- Flux de travail pour configurer une base de données source Oracle AWS autogérée ou gérée pour AWS DMS Configuration d'une base de données source Oracle
- Utilisation d'une base de données Oracle autogérée comme source pour AWS DMS
- Utilisation d'une base de données Oracle AWS gérée comme source pour AWS DMS
- Restrictions relatives à l'utilisation d'Oracle comme source pour AWS DMS
- Prise en charge de SSL pour un point de terminaison Oracle
- Méthodes de chiffrement prises en charge pour l'utilisation d'Oracle comme source pour AWS DMS
- Méthodes de compression prises en charge pour l'utilisation d'Oracle comme source pour AWS DMS
- Réplication de tables imbriquées en utilisant Oracle comme source pour AWS DMS
- Stockage de REDO sur Oracle ASM lors de l'utilisation d'Oracle comme source pour AWS DMS
- Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS
- Types de données sources pour Oracle

Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC

Dans AWS DMS, il existe deux méthodes pour lire les journaux de restauration lors de la capture des données de modification (CDC) pour Oracle en tant que source : Oracle LogMiner et AWS DMS Binary Reader. LogMiner est une API Oracle permettant de lire les journaux de journalisation en ligne et les fichiers de journalisation archivés. Le lecteur binaire est une AWS DMS méthode qui lit et analyse directement les fichiers de journalisation bruts. Ces méthodes présentent les fonctionnalités suivantes.

Fonctionnalité	LogMiner	Binary Reader
Facile à configurer	Oui	Non
Réduction de l'impact sur les E/S et le CPU du système source	Non	Oui
Meilleures performances de CDC	Non	Oui
Prend en charge les clusters de tables Oracle	Oui	Non

Fonctionnalité	LogMiner	Binary Reader
Prend en charge tous les types de compressions HCC Oracle	Oui	Partiellement Binary Reader ne prend pas en charge QUERY LOW pour les tâches avec CDC. Tous les autres types de HCC sont entièrement pris en charge.
Prise en charge des colonnes LOB dans Oracle 12c uniquement	Non (le support LOB n'est pas disponible LogMiner dans Oracle 12c.)	Oui
Prend en charge les instructions UPDATE qui n'affectent que les colonnes LOB	Non	Oui
Prend en charge le chiffrement transparent des données (TDE) Oracle	Partiellement Lorsque vous utilisez Oracle LogMiner, le chiffrement TDE au niveau des colonnes AWS DMS n'est pas pris en charge pour Amazon RDS for Oracle.	Partiellement Binary Reader prend en charge le chiffrement TDE uniquement pour les bases de données Oracle autogérées.

Fonctionnalité	LogMiner	Binary Reader
Prend en charge toutes les méthodes de compression Oracle	Oui	Non
Prend en charge les transactions XA	Non	Oui
RAC	Oui Non recommandé	Oui Fortement recommandé

> **Note**
>
> Par défaut, AWS DMS utilise Oracle LogMiner pour (CDC).
> AWS DMS prend en charge les méthodes de chiffrement transparent des données (TDE) lorsque vous travaillez avec une base de données source Oracle. Si les informations d'identification TDE que vous spécifiez sont incorrectes, la tâche de AWS DMS migration n'échoue pas, ce qui peut avoir un impact sur la réplication continue des tables chiffrées. Pour plus d'informations sur la spécification des informations d'identification TDE, consultez Méthodes de chiffrement prises en charge pour l'utilisation d'Oracle comme source pour AWS DMS.

Les principaux avantages de l'utilisation de LogMiner with AWS DMS sont les suivants :

- LogMiner prend en charge la plupart des options Oracle, telles que les options de chiffrement et de compression. Binary Reader ne prend pas en charge toutes les options Oracle, en particulier celles liées au chiffrement et à la compression.
- LogMiner offre une configuration plus simple, en particulier par rapport à la configuration à accès direct à Binary Reader ou lorsque les journaux de rétablissement sont gérés à l'aide d'Oracle Automatic Storage Management (ASM).
- LogMiner prend en charge les clusters de tables destinés à être utilisés par AWS DMS. Ce n'est pas le cas de Binary Reader.

Les principaux avantages de l'utilisation de Binary Reader AWS DMS sont les suivants :

- Pour les migrations impliquant un volume élevé de modifications, LogMiner cela peut avoir un impact sur les E/S ou le processeur de l'ordinateur hébergeant la base de données source Oracle. Binary Reader a moins de chances d'impacter les E/S ou le CPU, car les journaux sont extraits directement au lieu d'effectuer plusieurs requêtes sur la base de données.
- Pour les migrations impliquant un volume élevé de modifications, les performances du CDC sont généralement bien meilleures avec Binary Reader qu'avec Oracle LogMiner.
- Binary Reader prend en charge le CDC pour les LOB dans la version 12c d'Oracle. LogMinerne le fait pas.

En général, utilisez Oracle LogMiner pour migrer votre base de données Oracle, sauf si vous vous trouvez dans l'une des situations suivantes :

- Vous avez besoin d'exécuter plusieurs tâches de migration sur la base de données source Oracle.
- Le volume de modifications ou le volume de journaux redo sur la base de données Oracle source est élevé, ou vous avez des modifications et vous utilisez également Oracle ASM.

> **Note**
>
> Si vous passez d'Oracle LogMiner à AWS DMS Binary Reader, assurez-vous de redémarrer la tâche CDC.

Configuration de CDC sur une base de données source Oracle

Pour qu'un point de terminaison source Oracle puisse se connecter à la base de données pour une tâche de capture des données de modification (CDC), vous devrez peut-être spécifier des attributs de connexion supplémentaires. Vous devrez peut-être le faire pour une tâche de chargement complet et de CDC ou pour une tâche de CDC uniquement. Les attributs de connexion supplémentaires que vous spécifiez dépendent de la méthode que vous utilisez pour accéder aux journaux de journalisation : Oracle LogMiner ou AWS DMS Binary Reader.

Vous spécifiez des attributs de connexion supplémentaires lorsque vous créez un point de terminaison source. Si vous avez plusieurs paramètres d'attribut de connexion, séparez-les les uns des autres par des points-virgules, sans espace supplémentaire (par exemple, `oneSetting;thenAnother`).

AWS DMS utilise LogMiner par défaut. Vous n'avez pas à spécifier d'attributs de connexion supplémentaires pour l'utiliser.

Pour utiliser Binary Reader afin d'accéder aux journaux redo, ajoutez les attributs de connexion supplémentaires suivants.

```
useLogMinerReader=N;useBfile=Y;
```

Utilisez le format suivant pour les attributs de connexion supplémentaires afin d'accéder à un serveur qui utilise ASM avec Binary Reader.

```
useLogMinerReader=N;useBfile=Y;asm_user=asm_username;asm_server=RAC_server_ip_address:port_numb
+ASM;
```

Définissez le paramètre de demande `Password` du point de terminaison source sur le mot de passe d'utilisateur Oracle et le mot de passe ASM, séparés par une virgule comme suit.

```
oracle_user_password,asm_user_password
```

Dans les versions où la source Oracle utilise ASM, vous pouvez utiliser des options hautes performances dans Binary Reader pour le traitement des transactions à grande échelle. Ces options incluent des attributs de connexion supplémentaires pour spécifier le nombre de threads parallèles (`parallelASMReadThreads`) et le nombre de tampons de lecture anticipée (`readAheadBlocks`). La définition conjointe de ces attributs peut améliorer considérablement les performances de la tâche de CDC. Les paramètres suivants fournissent de bons résultats pour la plupart des configurations ASM.

```
useLogMinerReader=N;useBfile=Y;asm_user=asm_username;asm_server=RAC_server_ip_address:port_numb
+ASM;
    parallelASMReadThreads=6;readAheadBlocks=150000;
```

Pour de plus amples informations sur les valeurs prises en charge par les attributs de connexion supplémentaires, veuillez consulter Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS.

En outre, les performances d'une tâche de CDC avec une source Oracle qui utilise ASM dépendent d'autres paramètres que vous choisissez. Ces paramètres incluent vos attributs de connexion supplémentaires AWS DMS et les paramètres SQL pour configurer la source Oracle. Pour plus d'informations sur les attributs de connexion supplémentaires pour une source Oracle qui utilise ASM, consultez Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS.

Vous devez également choisir un point de départ de CDC approprié. Généralement, lorsque vous effectuez cette opération, vous souhaitez identifier le point de traitement des transactions qui capture la première transaction ouverte à partir de laquelle commencer la CDC. Dans le cas contraire, la tâche de CDC peut manquer les premières transactions ouvertes. Pour une base de données source Oracle, vous pouvez choisir un point de départ natif de CDC en fonction du numéro SCN Oracle afin d'identifier cette première transaction ouverte. Pour plus d'informations, consultez Exécution de la réplication à partir d'un point de départ CDC.

Pour plus d'informations sur la configuration de la CDC pour une base de données Oracle autogérée en tant que source, consultez Privilèges de compte requis lors de l'utilisation LogMiner d'Oracle pour accéder aux journaux de journalisation, Privilèges de compte requis lors de l'utilisation de AWS DMS Binary Reader pour accéder aux journaux de restauration et Privilèges de compte supplémentaires requis lors de l'utilisation de Binary Reader avec Oracle ASM.

Pour plus d'informations sur la configuration du CDC pour une base de données Oracle AWS gérée en tant que source, consultez Configuration d'une tâche CDC pour utiliser Binary Reader avec une source RDS pour Oracle pour AWS DMS etUtilisation d'Amazon RDS Oracle Standby (réplica de lecture) en tant que source avec Binary Reader pour la CDC dans AWS DMS.

Flux de travail pour configurer une base de données source Oracle AWS autogérée ou gérée pour AWS DMS

Flux de travail pour configurer une base de données source Oracle AWS autogérée ou gérée pour AWS DMS

Pour configurer une instance de base de données source autogérée, suivez les étapes du flux de travail suivantes, en fonction de la manière dont vous effectuez la CDC.

Pour cette étape du flux de travail	Si vous utilisez le CDC LogMiner, procédez comme suit	Si vous effectuez la CDC à l'aide de Binary Reader, procédez comme suit
Accordez des privilèges de compte Oracle.	veuillez consulter Privilèges de compte utilisateur requis sur une source Oracle autogérée pour AWS DMS.	veuillez consulter Privilèges de compte utilisateur requis sur une source Oracle autogérée pour AWS DMS.
Préparez la base de données source pour la réplication à l'aide de la CDC.	veuillez consulter Préparation d'une base de données source autogérée Oracle pour le CDC à l'aide de AWS DMS.	veuillez consulter Préparation d'une base de données source autogérée Oracle pour le CDC à l'aide de AWS DMS.
Accordez les privilèges d'utilisateur Oracle supplémentaires requis pour la CDC.	veuillez consulter Privilèges de compte requis lors de l'utilisation LogMiner d'Oracle pour accéder aux journaux de journalisation.	veuillez consulter Privilèges de compte requis lors de l'utilisation de AWS DMS Binary Reader pour accéder aux journaux de restauration.
Pour une instance Oracle avec ASM, accordez les privilèges de compte d'utilisateur supplémentaires requis pour accéder à ASM pour la CDC.	Aucune action supplémentaire. AWS DMS prend en charge Oracle ASM sans privilèges de compte supplémentaires.	veuillez consulter Privilèges de compte supplémentaires requis lors de l'utilisation de Binary Reader avec Oracle ASM.
Si ce n'est pas déjà fait, configurez la tâche à utiliser LogMiner ou Binary Reader for CDC.	veuillez consulter Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC.	veuillez consulter Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC.
Configurez Oracle Standby en tant que source pour la CDC.	AWS DMS ne prend pas en charge Oracle Standby en tant que source.	veuillez consulter Utilisation d'une instance autogérée d'Oracle Standby en tant que source avec Binary Reader pour la CDC dans AWS DMS.

Utilisez les étapes de flux de travail suivantes pour configurer une instance de base de données source Oracle AWS gérée.

Pour cette étape du flux de travail	Si vous utilisez le CDC LogMiner, procédez comme suit	Si vous effectuez la CDC à l'aide de Binary Reader, procédez comme suit
Accordez des privilèges de compte Oracle.	Pour plus d'informations, consultez Privilèges de compte utilisateur requis sur une source Oracle AWS gérée pour AWS DMS.	Pour plus d'informations, consultez Privilèges de compte utilisateur requis sur une source Oracle AWS gérée pour AWS DMS.
Préparez la base de données source pour la réplication à l'aide de la CDC.	Pour plus d'informations, consultez Configuration d'une source Oracle AWS gérée pour AWS DMS.	Pour plus d'informations, consultez Configuration d'une source Oracle AWS gérée pour AWS DMS.
Accordez les privilèges d'utilisateur Oracle supplémentaires requis pour la CDC.	Aucun privilège de compte supplémentaire n'est requis.	Pour plus d'informations, consultez Configuration d'une tâche CDC pour utiliser Binary Reader avec une source RDS pour Oracle pour AWS DMS.
Si ce n'est pas déjà fait, configurez la tâche à utiliser LogMiner ou Binary Reader for CDC.	Pour plus d'informations, consultez Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC.	Pour plus d'informations, consultez Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC.
Configurez Oracle Standby en tant que source pour la CDC.	AWS DMS ne prend pas en charge Oracle Standby en tant que source.	Pour plus d'informations, consultez Utilisation d'Amazon RDS Oracle Standby (réplica de lecture) en tant que source avec Binary Reader pour la CDC dans AWS DMS.

Utilisation d'une base de données Oracle autogérée comme source pour AWS DMS

Une base de données autogérée est une base de données que vous configurez et contrôlez. Il peut s'agir d'une instance de base de données locale sur site ou d'une base de données sur Amazon EC2. Vous trouverez ci-dessous des informations sur les privilèges et les configurations dont vous avez besoin pour utiliser une base de données Oracle autogérée avec AWS DMS.

Privilèges de compte utilisateur requis sur une source Oracle autogérée pour AWS DMS

Pour utiliser une base de données Oracle comme source dans AWS DMS, accordez les privilèges suivants à l'utilisateur Oracle spécifié dans les paramètres de connexion du point de terminaison Oracle.

> **Note**
>
> Lorsque vous accordez des privilèges, utilisez le nom réel des objets, et non leur synonyme. Par exemple, utilisez `V_$OBJECT` en intégrant le trait de soulignement, et non `V$OBJECT` sans le trait de soulignement.

```
GRANT CREATE SESSION TO db_user;
GRANT SELECT ANY TRANSACTION TO db_user;
GRANT SELECT ON V_$ARCHIVED_LOG TO db_user;
GRANT SELECT ON V_$LOG TO db_user;
GRANT SELECT ON V_$LOGFILE TO db_user;
GRANT SELECT ON V_$LOGMNR_LOGS TO db_user;
GRANT SELECT ON V_$LOGMNR_CONTENTS TO db_user;
GRANT SELECT ON V_$DATABASE TO db_user;
GRANT SELECT ON V_$THREAD TO db_user;
GRANT SELECT ON V_$PARAMETER TO db_user;
GRANT SELECT ON V_$NLS_PARAMETERS TO db_user;
GRANT SELECT ON V_$TIMEZONE_NAMES TO db_user;
GRANT SELECT ON V_$TRANSACTION TO db_user;
GRANT SELECT ON V_$CONTAINERS TO db_user;
GRANT SELECT ON ALL_INDEXES TO db_user;
GRANT SELECT ON ALL_OBJECTS TO db_user;
GRANT SELECT ON ALL_TABLES TO db_user;
GRANT SELECT ON ALL_USERS TO db_user;
GRANT SELECT ON ALL_CATALOG TO db_user;
GRANT SELECT ON ALL_CONSTRAINTS TO db_user;
GRANT SELECT ON ALL_CONS_COLUMNS TO db_user;
```

```
GRANT SELECT ON ALL_TAB_COLS TO db_user;
GRANT SELECT ON ALL_IND_COLUMNS TO db_user;
GRANT SELECT ON ALL_ENCRYPTED_COLUMNS TO db_user;
GRANT SELECT ON ALL_LOG_GROUPS TO db_user;
GRANT SELECT ON ALL_TAB_PARTITIONS TO db_user;
GRANT SELECT ON SYS.DBA_REGISTRY TO db_user;
GRANT SELECT ON SYS.OBJ$ TO db_user;
GRANT SELECT ON DBA_TABLESPACES TO db_user;
GRANT SELECT ON DBA_OBJECTS TO db_user; -- Required if the Oracle version is earlier
 than 11.2.0.3.
GRANT SELECT ON SYS.ENC$ TO db_user; -- Required if transparent data encryption (TDE)
 is enabled. For more information on using Oracle TDE with AWS DMS, see Méthodes de
 chiffrement prises en charge pour l'utilisation d'Oracle comme source pour AWS DMS.
GRANT SELECT ON GV_$TRANSACTION TO db_user; -- Required if the source database is
 Oracle RAC in AWS DMS versions 3.4.6 and higher.
GRANT SELECT ON V_$DATAGUARD_STATS TO db_user; -- Required if the source database is
 Oracle Data Guard and Oracle Standby is used in the latest release of DMS version
 3.4.6, version 3.4.7, and higher.
```

Accordez le privilège supplémentaire suivant pour chaque table répliquée lorsque vous utilisez une liste de tables spécifique.

```
GRANT SELECT on any-replicated-table to db_user;
```

Accordez le privilège supplémentaire suivant pour valider les colonnes LOB avec la fonctionnalité de validation.

```
GRANT EXECUTE ON SYS.DBMS_CRYPTO TO db_user;
```

Accordez le privilège supplémentaire suivant si vous utilisez un lecteur binaire au lieu de LogMiner.

```
GRANT SELECT ON SYS.DBA_DIRECTORIES TO db_user;
```

Accordez le privilège supplémentaire suivant pour exposer les vues.

```
GRANT SELECT on ALL_VIEWS to dms_user;
```

Pour exposer les vues, vous devez également ajouter l'attribut de connexion supplémentaire `exposeViews=true` au point de terminaison source.

Accordez le privilège supplémentaire suivant lorsque vous utilisez des réplications sans serveur.

```
GRANT SELECT on dba_segments to db_user;
```

Pour en savoir plus sur les réplications sans serveur, consultez Travailler avec AWS DMS Serverless.

Accordez les privilèges supplémentaires suivants lorsque vous utilisez des évaluations de prémigration spécifiques à Oracle.

```
GRANT SELECT on gv_$parameter  to dms_user;
GRANT SELECT on v_$instance to dms_user;
GRANT SELECT on v_$version to dms_user;
GRANT SELECT on gv_$ASM_DISKGROUP to dms_user;
GRANT SELECT on gv_$database to dms_user;
GRANT SELECT on dba_db_links to dms_user;
GRANT SELECT on gv_$log_History to dms_user;
GRANT SELECT on gv_$log to dms_user;
GRANT SELECT ON DBA_TYPES TO db_user;
```

Pour en savoir plus sur les évaluations de prémigration spécifiques à Oracle, consultez Évaluations Oracle.

Conditions préalables à la gestion des transactions ouvertes pour Oracle Standby

Lorsque vous utilisez AWS DMS les versions 3.4.6 ou supérieures, effectuez les étapes suivantes pour gérer les transactions ouvertes pour Oracle Standby.

1. Créez un lien de base de données nommé `AWSDMS_DBLINK` sur la base de données principale. *DMS_USER* utilisera ce lien pour se connecter à la base de données principale. Notez que le lien de base de données est exécuté à partir de l'instance de secours pour interroger les transactions ouvertes exécutées sur la base de données principale. Consultez l'exemple suivant.

   ```
   CREATE PUBLIC DATABASE LINK AWSDMS_DBLINK
      CONNECT TO DMS_USER IDENTIFIED BY DMS_USER_PASSWORD
      USING '(DESCRIPTION=
               (ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_HOST_NAME_OR_IP)(PORT=PORT))
   ```

```
            (CONNECT_DATA=(SERVICE_NAME=SID))
          )';
```

2. Vérifiez que la connexion au lien de base de données avec *DMS_USER* est établie, comme indiqué dans l'exemple suivant.

```
select 1 from dual@AWSDMS_DBLINK
```

Préparation d'une base de données source autogérée Oracle pour le CDC à l'aide de AWS DMS

Préparez la base de données Oracle autogérée en tant que source pour exécuter une tâche de CDC en procédant comme suit :

- Vérifier que la version de la base de données source est prise en AWS DMS charge.
- Vérifiez que le mode ARCHIVELOG est activé.
- Configurez une journalisation supplémentaire.

Vérifier que la version de la base de données source est prise en AWS DMS charge

Exécutez une requête similaire à ce qui suit pour vérifier que la version actuelle de la base de données source Oracle est prise en charge par AWS DMS.

```
SELECT name, value, description FROM v$parameter WHERE name = 'compatible';
```

Ici, `name`, `value` et `description` sont des colonnes quelque part dans la base de données qui sont interrogées en fonction de la valeur de `name`. Si cette requête s'exécute sans erreur, elle AWS DMS prend en charge la version actuelle de la base de données et vous pouvez poursuivre la migration. Si la requête génère une erreur, AWS DMS cela signifie que la version actuelle de la base de données n'est pas prise en charge. Pour procéder à la migration, convertissez d'abord la base de données Oracle vers une version prise en charge par AWS DMS.

Vérifiez que le mode ARCHIVELOG est activé

Vous pouvez exécuter Oracle en deux modes différents : le mode `ARCHIVELOG` et le mode `NOARCHIVELOG`. Pour exécuter une tâche de CDC, exécutez la base de données en mode `ARCHIVELOG`. Pour savoir si la base de données est en mode `ARCHIVELOG`, exécutez la requête suivante.

```
SQL> SELECT log_mode FROM v$database;
```

Si le mode NOARCHIVELOG est renvoyé, définissez la base de données sur ARCHIVELOG conformément aux instructions d'Oracle.

Configurez une journalisation supplémentaire

Pour enregistrer les modifications en cours, AWS DMS vous devez activer une journalisation supplémentaire minimale sur votre base de données source Oracle. En outre, vous devez activer la journalisation supplémentaire sur chaque table répliquée de la base de données.

Par défaut, AWS DMS ajoute une journalisation PRIMARY KEY supplémentaire sur toutes les tables répliquées. AWS DMS Pour autoriser l'ajout d'une journalisation PRIMARY KEY supplémentaire, accordez le privilège suivant pour chaque table répliquée.

```
ALTER on any-replicated-table;
```

Vous pouvez désactiver la journalisation PRIMARY KEY supplémentaire par défaut ajoutée en AWS DMS utilisant l'attribut addSupplementalLogging de connexion supplémentaire. Pour plus d'informations, consultez Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS.

Assurez-vous d'activer la journalisation supplémentaire si votre tâche de réplication met à jour une table en utilisant une clause WHERE qui ne fait pas référence à une colonne de clé primaire.

Pour configurer manuellement la journalisation supplémentaire

1. Exécutez la requête suivante pour vérifier si la journalisation supplémentaire est déjà activée pour la base de données.

   ```
   SELECT supplemental_log_data_min FROM v$database;
   ```

   Si le résultat renvoyé est YES ou IMPLICIT, la journalisation supplémentaire est activée pour la base de données.

   Sinon, activez la journalisation supplémentaire pour la base de données en exécutant la commande suivante.

   ```
   ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
   ```

2. Assurez-vous que la journalisation supplémentaire requise est ajoutée pour chaque table répliquée.

   Éléments à prendre en compte :

   - Si la journalisation supplémentaire `ALL COLUMNS` est ajoutée à la table, il n'est pas nécessaire d'en ajouter une autre.
   - Si une clé primaire existe, ajoutez une journalisation supplémentaire pour la clé primaire. Vous pouvez le faire en utilisant le format pour ajouter une journalisation supplémentaire sur la clé primaire en elle-même, ou en ajoutant une journalisation supplémentaire sur les colonnes de clé primaire dans la base de données.

     ```
     ALTER TABLE Tablename ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
     ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
     ```

   - Si aucune clé primaire n'existe et que la table possède un seul index unique, toutes les colonnes de l'index unique doivent être ajoutées au journal supplémentaire.

     ```
     ALTER TABLE TableName ADD SUPPLEMENTAL LOG GROUP LogGroupName
      (UniqueIndexColumn1[, UniqueIndexColumn2] ...) ALWAYS;
     ```

     L'utilisation de la commande `SUPPLEMENTAL LOG DATA (UNIQUE INDEX) COLUMNS` n'ajoute pas les colonnes d'index uniques dans le journal.

   - S'il n'existe aucune clé primaire et que la table comporte plusieurs index uniques, AWS DMS sélectionne le premier index unique dans une liste alphabétique croissante. Vous devez ajouter une journalisation supplémentaire sur les colonnes de l'index sélectionné, comme au point précédent.

     L'utilisation de la commande `SUPPLEMENTAL LOG DATA (UNIQUE INDEX) COLUMNS` n'ajoute pas les colonnes d'index uniques dans le journal.

   - Si aucune clé primaire n'existe et qu'il n'y a pas d'index unique, ajoutez une journalisation supplémentaire sur toutes les colonnes.

     ```
     ALTER TABLE TableName ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
     ```

     Dans certains cas, l'index unique et la clé primaire de la table cible sont différents de ceux de la table source. Dans ce cas, ajoutez manuellement une journalisation supplémentaire sur les colonnes de la table source qui constituent l'index unique ou la clé primaire de la table cible.

De plus, si vous modifiez la clé primaire de la table cible, vous devez ajouter la journalisation supplémentaire dans les colonnes de l'index cible, au lieu des colonnes de la clé primaire ou de l'index unique source.

Si un filtre ou une transformation est défini pour une table, vous devrez peut-être activer la journalisation supplémentaire.

Éléments à prendre en compte :

- Si la journalisation supplémentaire `ALL COLUMNS` est ajoutée à la table, il n'est pas nécessaire d'en ajouter une autre.
- Si la table a un index unique ou une clé primaire, ajoutez une journalisation supplémentaire sur chaque colonne impliquée dans un filtre ou une transformation. Toutefois, ne le faites que si ces colonnes sont différentes des colonnes de la clé primaire ou de l'index unique.
- Si une transformation inclut une seule colonne, n'ajoutez pas cette colonne à un groupe de journalisations supplémentaires. Par exemple, pour une transformation `A+B`, ajoutez une journalisation supplémentaire sur les deux colonnes `A` et `B`. Cependant, pour une transformation `substring(A,10)`, n'ajoutez pas de journalisation supplémentaire sur la colonne `A`.
- Pour configurer une journalisation supplémentaire sur les colonnes de clé primaire ou d'index unique et sur d'autres colonnes filtrées ou transformées, vous pouvez configurer la journalisation supplémentaire `USER_LOG_GROUP`. Ajoutez cette journalisation à la fois sur les colonnes de clé primaire ou d'index unique et sur toute autre colonne spécifique filtrée ou transformée.

  Par exemple, pour répliquer une table nommée `TEST.LOGGING` avec une clé primaire `ID` et un filtre sur la colonne `NAME`, vous pouvez exécuter une commande similaire à celle qui suit pour créer la journalisation supplémentaire du groupe de journaux.

```
ALTER TABLE TEST.LOGGING ADD SUPPLEMENTAL LOG GROUP TEST_LOG_GROUP (ID, NAME) ALWAYS;
```

Privilèges de compte requis lors de l'utilisation LogMiner d'Oracle pour accéder aux journaux de journalisation

Pour accéder aux journaux de journalisation à l'aide d'Oracle LogMiner, accordez les privilèges suivants à l'utilisateur Oracle spécifié dans les paramètres de connexion du point de terminaison Oracle.

```
GRANT EXECUTE on DBMS_LOGMNR to db_user;
GRANT SELECT on V_$LOGMNR_LOGS to db_user;
GRANT SELECT on V_$LOGMNR_CONTENTS to db_user;
GRANT LOGMINING to db_user; -- Required only if the Oracle version is 12c or higher.
```

Privilèges de compte requis lors de l'utilisation de AWS DMS Binary Reader pour accéder aux journaux de restauration

Pour accéder aux journaux de journalisation à l'aide du lecteur AWS DMS binaire, accordez les privilèges suivants à l'utilisateur Oracle spécifié dans les paramètres de connexion du point de terminaison Oracle.

```
GRANT SELECT on v_$transportable_platform to db_user;   -- Grant this privilege if the
 redo logs are stored in Oracle Automatic Storage Management (ASM) and AWS DMS accesses
 them from ASM.
GRANT CREATE ANY DIRECTORY to db_user;                  -- Grant this privilege to
 allow AWS DMS to use Oracle BFILE read file access in certain cases. This access is
 required when the replication instance doesn't have file-level access to the redo logs
 and the redo logs are on non-ASM storage.
GRANT EXECUTE on DBMS_FILE_TRANSFER to db_user;         -- Grant this privilege to copy
 the redo log files to a temporary folder using the CopyToTempFolder method.
GRANT EXECUTE on DBMS_FILE_GROUP to db_user;
```

Binary Reader fonctionne avec les fonctions de fichier Oracle qui incluent des répertoires Oracle. Chaque objet répertoire Oracle inclut le nom du dossier contenant les journaux redo à traiter. Ces répertoires Oracle ne sont pas représentés au niveau du système de fichiers. Il s'agit plutôt de répertoires logiques créés au niveau de la base de données Oracle. Vous pouvez les afficher dans la vue Oracle `ALL_DIRECTORIES`.

Si vous souhaitez AWS DMS créer ces annuaires Oracle, accordez le `CREATE ANY DIRECTORY` privilège spécifié ci-dessus. AWS DMS crée les noms de répertoires avec le `DMS_` préfixe. Si vous n'accordez pas le privilège `CREATE ANY DIRECTORY`, créez manuellement les répertoires correspondants. Dans certains cas, lorsque vous créez manuellement les répertoires Oracle, l'utilisateur Oracle spécifié dans le point de terminaison source Oracle n'est pas l'utilisateur qui a créé ces répertoires. Dans ces cas, accordez également le privilège `READ on DIRECTORY`.

Si le point de terminaison source Oracle est en mode Active Dataguard Standby (ADG), consultez le billet Comment utiliser le lecteur binaire avec ADG sur le blog de base de données. AWS

> **Note**
>
> AWS DMS Le CDC ne prend pas en charge Active Dataguard Standby qui n'est pas configuré pour utiliser le service de rétablissement automatique du transport.

Dans certains cas, vous pouvez utiliser Oracle Managed Files (OMF) pour stocker les journaux. Ou le point de terminaison source se trouve dans ADG et vous ne pouvez donc pas accorder le privilège CREATE ANY DIRECTORY. Dans ces cas, créez manuellement les répertoires avec tous les emplacements de journal possibles avant de démarrer la tâche de AWS DMS réplication. Si AWS DMS ne trouve pas le répertoire précréé qu'il attend, la tâche s'arrête. En outre, AWS DMS ne supprime pas les entrées qu'il a créées dans la vue `ALL_DIRECTORIES`, donc supprimez-les manuellement.

Privilèges de compte supplémentaires requis lors de l'utilisation de Binary Reader avec Oracle ASM

Pour accéder aux journaux redo dans Automatic Storage Management (ASM) à l'aide de Binary Reader, accordez les privilèges suivants à l'utilisateur Oracle spécifié dans les paramètres de connexion au point de terminaison Oracle.

```
SELECT ON v_$transportable_platform
SYSASM -- To access the ASM account with Oracle 11g Release 2 (version 11.2.0.2) and
 higher, grant the Oracle endpoint user the SYSASM privilege. For older supported
 Oracle versions, it's typically sufficient to grant the Oracle endpoint user the
 SYSDBA privilege.
```

Vous pouvez valider l'accès au compte ASM en ouvrant une invite de commandes et en invoquant l'une des instructions suivantes, en fonction de votre version d'Oracle spécifiée précédemment.

Si vous avez besoin du privilège `SYSDBA`, utilisez ce qui suit.

```
sqlplus asmuser/asmpassword@+asmserver as sysdba
```

Si vous avez besoin du privilège `SYSASM`, utilisez ce qui suit.

```
sqlplus asmuser/asmpassword@+asmserver as sysasm
```

Utilisation d'une instance autogérée d'Oracle Standby en tant que source avec Binary Reader pour la CDC dans AWS DMS

Pour configurer une instance d'Oracle Standby en tant que source lors de l'utilisation de Binary Reader pour la CDC, commencez par prendre connaissance des prérequis suivants :

- AWS DMS prend actuellement en charge uniquement Oracle Active Data Guard Standby.
- Assurez-vous que la configuration d'Oracle Data Guard utilise :
  - Services de transport redo pour les transferts automatisés de données redo.
  - Services d'application pour appliquer automatiquement redo à la base de données de secours.

Pour confirmer que ces exigences sont satisfaites, exécutez la requête suivante.

```
SQL> select open_mode, database_role from v$database;
```

À partir du résultat de cette requête, vérifiez que la base de données de secours est ouverte en mode READ ONLY et que redo est appliqué automatiquement. Par exemple :

```
OPEN_MODE            DATABASE_ROLE
--------------------  ----------------
READ ONLY WITH APPLY  PHYSICAL STANDBY
```

Pour configurer une instance d'Oracle Standby en tant que source lors de l'utilisation de Binary Reader pour la CDC

1. Accordez les privilèges supplémentaires requis pour accéder aux fichiers journaux de secours.

   ```
   GRANT SELECT ON v_$standby_log TO db_user;
   ```

2. Créez un point de terminaison source pour Oracle Standby à l'aide d' AWS Management Console ou d' AWS CLI. Lors de la création du point de terminaison, spécifiez les attributs de connexion supplémentaires suivants.

   ```
   useLogminerReader=N;useBfile=Y;
   ```

> **Note**
>
> Dans AWS DMS, vous pouvez utiliser des attributs de connexion supplémentaires pour spécifier si vous souhaitez migrer à partir des journaux d'archives plutôt que des journaux de rétablissement. Pour plus d'informations, consultez Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS.

3. Configurez la destination du journal archivé.

   DMS Binary Reader pour la source Oracle sans ASM utilise les répertoires Oracle pour accéder aux journaux redo archivés. Si la base de données est configurée de sorte à utiliser la zone de récupération rapide (FRA) comme destination du journal d'archivage, l'emplacement des fichiers redo d'archivage n'est pas constant. Chaque jour de génération de journaux redo archivés entraîne la création d'un nouveau répertoire dans la FRA en utilisant le format de nom de répertoire AAAA_MM_JJ. Par exemple :

   ```
   DB_RECOVERY_FILE_DEST/SID/archivelog/YYYY_MM_DD
   ```

   Lorsque DMS a besoin d'accéder à des fichiers redo archivés dans le répertoire FRA récemment créé et que la base de données principale en lecture-écriture est utilisée en tant que source, DMS crée un nouveau répertoire Oracle ou remplace un répertoire Oracle existant, comme suit.

   ```
   CREATE OR REPLACE DIRECTORY dmsrep_taskid AS 'DB_RECOVERY_FILE_DEST/SID/archivelog/
   YYYY_MM_DD';
   ```

   Lorsque la base de données de secours est utilisée en tant que source, DMS ne parvient pas à créer ou à remplacer le répertoire Oracle, car la base de données est en mode lecture seule. Toutefois, vous pouvez choisir d'effectuer l'une des étapes supplémentaires suivantes :

   a. Modifiez `log_archive_dest_id_1` de sorte à utiliser un chemin réel plutôt que la FRA dans une configuration de ce type où Oracle ne créera pas de sous-répertoires quotidiens :

      ```
      ALTER SYSTEM SET log_archive_dest_1='LOCATION=full directory path'
      ```

      Créez ensuite un objet de répertoire Oracle à utiliser par DMS :

      ```
      CREATE OR REPLACE DIRECTORY dms_archived_logs AS 'full directory path';
      ```

b. Créez une destination de journal d'archivage supplémentaire et un objet de répertoire Oracle pointant vers cette destination. Par exemple :

```
ALTER SYSTEM SET log_archive_dest_3='LOCATION=full directory path';
CREATE DIRECTORY dms_archived_log AS 'full directory path';
```

Ajoutez ensuite un attribut de connexion supplémentaire au point de terminaison source de la tâche :

```
archivedLogDestId=3
```

c. Précréez manuellement des objets de répertoire Oracle à utiliser par DMS.

```
CREATE DIRECTORY dms_archived_log_20210301 AS 'DB_RECOVERY_FILE_DEST/SID/
archivelog/2021_03_01';
CREATE DIRECTORY dms_archived_log_20210302 AS 'DB_RECOVERY_FILE_DEST>/SID>/
archivelog/2021_03_02';
...
```

d. Créez une tâche de planificateur Oracle qui s'exécute quotidiennement et crée le répertoire requis.

Utilisation d'une base de données gérée par l'utilisateur sur Oracle Cloud Infrastructure (OCI) en tant que source pour la CDC dans AWS DMS

Une base de données gérée par l'utilisateur est une base de données que vous configurez et contrôlez, comme une base de données Oracle créée sur une machine virtuelle (VM), un matériel nu ou un serveur Exadata. Il peut également s'agir de bases de données que vous configurez et contrôlez et qui s'exécutent sur une infrastructure dédiée, comme Oracle Cloud Infrastructure (OCI). Les informations suivantes décrivent les privilèges et les configurations dont vous avez besoin lorsque vous utilisez une base de données Oracle gérée par l'utilisateur sur OCI en tant que source pour la capture des données de modification (CDC) dans AWS DMS.

Pour configurer une base de données Oracle gérée par l'utilisateur et hébergée par OCI en tant que source pour la capture des données de modification

1. Accordez les privilèges de compte d'utilisateur requis pour une base de données source Oracle gérée par l'utilisateur sur OCI. Pour plus d'informations, consultez Privilèges de compte pour un point de terminaison source Oracle autogéré.

2. Accordez les privilèges de compte requis lors de l'utilisation de Binary Reader pour accéder aux journaux redo. Pour plus d'informations, consultez Privilèges de compte requis lors de l'utilisation de Binary Reader.
3. Ajoutez les privilèges de compte requis lors de l'utilisation de Binary Reader avec Oracle Automatic Storage Management (ASM). Pour plus d'informations, consultez Privilèges de compte supplémentaires requis lors de l'utilisation de Binary Reader avec Oracle ASM.
4. Configurez une journalisation supplémentaire. Pour plus d'informations, consultez Configuration d'une journalisation supplémentaire.
5. Configurez le chiffrement TDE. Pour plus d'informations, consultez Méthodes de chiffrement lors de l'utilisation d'une base de données Oracle comme point de terminaison source.

Les limitations suivantes s'appliquent lors de la réplication de données à partir d'une base de données source Oracle sur Oracle Cloud Infrastructure (OCI).

Limites

- DMS ne prend pas en charge l'utilisation d'Oracle LogMiner pour accéder aux journaux de journalisation.
- DMS ne prend pas en charge Autonomous DB.

Utilisation d'une base de données Oracle AWS gérée comme source pour AWS DMS

Une base de données AWS gérée est une base de données qui se trouve sur un service Amazon tel qu'Amazon RDS, Amazon Aurora ou Amazon S3. Vous trouverez ci-dessous les privilèges et les configurations que vous devez configurer lorsque vous utilisez une base de données Oracle AWS gérée avec AWS DMS.

Privilèges de compte utilisateur requis sur une source Oracle AWS gérée pour AWS DMS

Accordez les privilèges suivants au compte d'utilisateur Oracle spécifié dans la définition du point de terminaison source Oracle.

> **Important**
>
> Pour toutes les valeurs de paramètre telles que *db_user* et *any-replicated-table*, Oracle suppose que la valeur est entièrement en majuscules, sauf si vous spécifiez la valeur avec un identifiant sensible à la casse. Supposons, par exemple, que vous créez une valeur *db_user* sans utiliser de guillemets, comme dans `CREATE USER` *myuser* ou `CREATE`

USER MYUSER. Dans ce cas, Oracle identifie et stocke la valeur entièrement en majuscules (MYUSER). Si vous utilisez des guillemets, comme dans CREATE USER "MyUser" ou CREATE USER 'MyUser', Oracle identifie et stocke la valeur sensible à la casse que vous spécifiez (MyUser).

```
GRANT CREATE SESSION to db_user;
GRANT SELECT ANY TRANSACTION to db_user;
GRANT SELECT on DBA_TABLESPACES to db_user;
GRANT SELECT ON any-replicated-table to db_user;
GRANT EXECUTE on rdsadmin.rdsadmin_util to db_user;
 -- For Oracle 12c or higher:
GRANT LOGMINING to db_user; - Required only if the Oracle version is 12c or higher.
```

De plus, accordez les autorisations SELECT et EXECUTE sur les objets SYS à l'aide de la procédure Amazon RDS rdsadmin.rdsadmin_util.grant_sys_object, comme indiqué. Pour plus d'informations, consultez la page Attribution des privilèges SELECT ou EXECUTE aux objets SYS.

```
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_VIEWS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_TAB_PARTITIONS', 'db_user',
 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_INDEXES', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_OBJECTS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_TABLES', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_USERS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_CATALOG', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_CONSTRAINTS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_CONS_COLUMNS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_TAB_COLS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_IND_COLUMNS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_LOG_GROUPS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$ARCHIVED_LOG', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$LOG', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$LOGFILE', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$DATABASE', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$THREAD', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$PARAMETER', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$NLS_PARAMETERS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$TIMEZONE_NAMES', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$TRANSACTION', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$CONTAINERS', 'db_user', 'SELECT');
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_REGISTRY', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('OBJ$', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_ENCRYPTED_COLUMNS', 'db_user',
 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$LOGMNR_LOGS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$LOGMNR_CONTENTS','db_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_LOGMNR', 'db_user', 'EXECUTE');

-- (as of Oracle versions 12.1 and higher)
exec rdsadmin.rdsadmin_util.grant_sys_object('REGISTRY$SQLPATCH', 'db_user', 'SELECT');

-- (for Amazon RDS Active Dataguard Standby (ADG))
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$STANDBY_LOG', 'db_user', 'SELECT');

-- (for transparent data encryption (TDE))

exec rdsadmin.rdsadmin_util.grant_sys_object('ENC$', 'db_user', 'SELECT');

-- (for validation with LOB columns)
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_CRYPTO', 'db_user', 'EXECUTE');

-- (for binary reader)
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_DIRECTORIES','db_user','SELECT');

-- Required when the source database is Oracle Data guard, and Oracle Standby is used
 in the latest release of DMS version 3.4.6, version 3.4.7, and higher.

exec rdsadmin.rdsadmin_util.grant_sys_object('V_$DATAGUARD_STATS', 'db_user',
 'SELECT');
```

Pour plus d'informations sur l'utilisation d'Amazon RDS Active Dataguard Standby (ADG) avec AWS DMS , consultez Utilisation d'Amazon RDS Oracle Standby (réplica de lecture) en tant que source avec Binary Reader pour la CDC dans AWS DMS.

Pour plus d'informations sur l'utilisation d'Oracle TDE avec AWS DMS, consultezMéthodes de chiffrement prises en charge pour l'utilisation d'Oracle comme source pour AWS DMS.

Conditions préalables à la gestion des transactions ouvertes pour Oracle Standby

Lorsque vous utilisez AWS DMS les versions 3.4.6 ou supérieures, effectuez les étapes suivantes pour gérer les transactions ouvertes pour Oracle Standby.

1. Créez un lien de base de données nommé AWSDMS_DBLINK sur la base de données principale. *DMS_USER* utilisera ce lien pour se connecter à la base de données principale. Notez que le lien de base de données est exécuté à partir de l'instance de secours pour interroger les transactions ouvertes exécutées sur la base de données principale. Consultez l'exemple suivant.

```
CREATE PUBLIC DATABASE LINK AWSDMS_DBLINK
   CONNECT TO DMS_USER IDENTIFIED BY DMS_USER_PASSWORD
   USING '(DESCRIPTION=
            (ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_HOST_NAME_OR_IP)(PORT=PORT))
            (CONNECT_DATA=(SERVICE_NAME=SID))
          )';
```

2. Vérifiez que la connexion au lien de base de données avec *DMS_USER* est établie, comme indiqué dans l'exemple suivant.

```
select 1 from dual@AWSDMS_DBLINK
```

Configuration d'une source Oracle AWS gérée pour AWS DMS

Avant d'utiliser une base de données Oracle AWS gérée comme source pour AWS DMS, effectuez les tâches suivantes pour la base de données Oracle :

- Activez les sauvegardes automatiques. Pour plus d'informations sur l'activation des sauvegardes automatiques, consultez Activation des sauvegardes automatiques dans le Guide de l'utilisateur Amazon RDS.
- Configurez la journalisation supplémentaire.
- Configurez l'archivage. L'archivage des journaux redo de votre instance de base de données Amazon RDS for Oracle permet de récupérer les informations du journal AWS DMS à l'aide d'Oracle LogMiner ou de Binary Reader.

Pour configurer l'archivage

1. Exécutez la commande `rdsadmin.rdsadmin_util.set_configuration` pour configurer l'archivage.

   Par exemple, pour conserver les journaux redo archivés pendant 24 heures, exécutez la commande suivante.

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24);
commit;
```

> **Note**
>
> La validation est obligatoire pour qu'une modification prenne effet.

2. Vérifiez que votre espace de stockage est suffisant pour les journaux redo archivés pendant la période spécifiée. Par exemple, si votre période de conservation est de 24 heures, calculez la taille totale de vos journaux redo archivés accumulés sur une heure type de traitement des transactions et multipliez ce total par 24. Comparez ce total calculé sur 24 heures avec votre espace de stockage disponible et déterminez si vous disposez de suffisamment d'espace de stockage pour traiter les transactions 24 h/24.

Pour configurer la journalisation supplémentaire

1. Exécutez la commande suivante pour activer la journalisation supplémentaire au niveau de la base de données.

```
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD');
```

2. Exécutez la commande suivante pour activer la journalisation supplémentaire des clés primaires.

```
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD','PRIMARY KEY');
```

3. (Facultatif) Activez la journalisation supplémentaire au niveau des clés, au niveau de la table.

   Votre base de données source exige un peu plus de temps système lorsqu'une journalisation supplémentaire au niveau des clés est activée. Par conséquent, si vous ne migrez qu'un sous-ensemble de vos tables, vous souhaiterez peut-être activer la journalisation supplémentaire au niveau des clés, au niveau de la table. Pour activer la journalisation supplémentaire au niveau des clés, au niveau de la table, exécutez la commande suivante.

```
alter table table_name add supplemental log data (PRIMARY KEY) columns;
```

Configuration d'une tâche CDC pour utiliser Binary Reader avec une source RDS pour Oracle pour AWS DMS

Vous pouvez configurer l'accès AWS DMS à la source des journaux redo de l'instance Amazon RDS pour Oracle à l'aide de Binary Reader for CDC.

> **Note**
>
> Pour utiliser Oracle LogMiner, les privilèges de compte utilisateur minimaux requis sont suffisants. Pour plus d'informations, consultez Privilèges de compte utilisateur requis sur une source Oracle AWS gérée pour AWS DMS.

Pour utiliser AWS DMS Binary Reader, spécifiez des paramètres supplémentaires et des attributs de connexion supplémentaires pour le point de terminaison source Oracle, en fonction de votre AWS DMS version.

La prise en charge de Binary Reader est disponible dans les versions suivantes d'Amazon RDS for Oracle :

- Oracle 11.2 : versions 11.2.0.4V11 et ultérieures
- Oracle 12.1 : versions 12.1.0.2.V7 et ultérieures
- Oracle 12.2 : toutes les versions
- Oracle 18.0 : toutes les versions
- Oracle 19.0 : toutes les versions

Pour configurer CDC à l'aide d' Binary Reader

1. Connectez-vous à la base de données source Amazon RDS for Oracle en tant qu'utilisateur principal et exécutez les procédures stockées suivantes pour créer les répertoires au niveau du serveur.

   ```
   exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;
   exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
   ```

2. Accordez les privilèges suivants au compte d'utilisateur Oracle utilisé pour accéder au point de terminaison source Oracle.

```
GRANT READ ON DIRECTORY ONLINELOG_DIR TO db_user;
GRANT READ ON DIRECTORY ARCHIVELOG_DIR TO db_user;
```

3. Définissez les attributs de connexion supplémentaires suivants sur le point de terminaison source Amazon RDS Oracle :

   - Pour RDS Oracle versions 11.2 et 12.1, définissez les attributs suivants.

     ```
     useLogminerReader=N;useBfile=Y;accessAlternateDirectly=false;useAlternateFolderForOnline=
     oraclePathPrefix=/rdsdbdata/db/{$DATABASE_NAME}_A/;usePathPrefix=/rdsdbdata/
     log/;replacePathPrefix=true;
     ```

   - Pour RDS Oracle versions 12.2, 18.0 et 19.0, définissez les attributs suivants.

     ```
     useLogminerReader=N;useBfile=Y;
     ```

> **Note**
>
> Assurez-vous qu'il n'y ait pas d'espace après le point-virgule (;) si vous avez plusieurs paramètres d'attribut ; par exemple, `oneSetting;thenAnother`.

Pour plus d'informations sur la configuration d'une tâche de CDC, consultez Configuration de CDC sur une base de données source Oracle.

Utilisation d'Amazon RDS Oracle Standby (réplica de lecture) en tant que source avec Binary Reader pour la CDC dans AWS DMS

Vérifiez les conditions préalables suivantes pour utiliser Amazon RDS for Oracle Standby en tant que source lors de l'utilisation de Binary Reader pour la CDC dans AWS DMS :

- Utilisez l'utilisateur principal Oracle pour configurer Binary Reader.
- Assurez-vous qu'il ne prend AWS DMS actuellement en charge que l'utilisation d'Oracle Active Data Guard Standby.

Ensuite, procédez comme suit pour utiliser RDS for Oracle Standby en tant que source lors de l'utilisation de Binary Reader pour la CDC.

Pour configurer RDS for Oracle Standby en tant que source lors de l'utilisation de Binary Reader pour la CDC

1. Connectez-vous à l'instance principale de RDS for Oracle en tant qu'utilisateur principal.
2. Exécutez les procédures stockées suivantes, comme indiqué dans le Guide de l'utilisateur Amazon RDS, pour créer les répertoires au niveau du serveur.

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

3. Identifiez les répertoires créés à l'étape 2.

```
SELECT directory_name, directory_path FROM all_directories
WHERE directory_name LIKE ( 'ARCHIVELOG_DIR_%' )
        OR directory_name LIKE ( 'ONLINELOG_DIR_%' )
```

   Par exemple, le code précédent affiche une liste de répertoires comme ce qui suit.

DIRECTORY_NAME	DIRECTORY_PATH
ARCHIVELOG_DIR_A	/rdsdbdata/db/ORCL_A/arch
ARCHIVELOG_DIR_B	/rdsdbdata/db/ORCL_B/arch
ONLINELOG_DIR_A	/rdsdbdata/db/ORCL_A/onlinelog
ONLINELOG_DIR_B	/rdsdbdata/db/ORCL_B/onlinelog

4. Accordez le privilège `Read` sur les répertoires précédents au compte d'utilisateur Oracle utilisé pour accéder à Oracle Standby.

```
GRANT READ ON DIRECTORY ARCHIVELOG_DIR_A TO db_user;
GRANT READ ON DIRECTORY ARCHIVELOG_DIR_B TO db_user;
GRANT READ ON DIRECTORY ONLINELOG_DIR_A TO db_user;
GRANT READ ON DIRECTORY ONLINELOG_DIR_B TO db_user;
```

5. Changez de journal d'archivage sur l'instance principale. Cela garantit que les modifications apportées à `ALL_DIRECTORIES` sont également transmises à Oracle Standby.
6. Exécutez une requête `ALL_DIRECTORIES` sur Oracle Standby pour confirmer que les modifications ont été appliquées.

7. Créez un point de terminaison source pour Oracle Standby à l'aide de la console AWS DMS de gestion ou AWS Command Line Interface (AWS CLI). Lors de la création du point de terminaison, spécifiez les attributs de connexion supplémentaires suivants.

```
useLogminerReader=N;useBfile=Y;archivedLogDestId=1;additionalArchivedLogDestId=2
```

8. Après avoir créé le point de terminaison, utilisez Tester la connexion du point de terminaison sur la page Créer un point de terminaison de la console ou utilisez la AWS CLI `test-connection` commande pour vérifier que la connectivité est établie.

Restrictions relatives à l'utilisation d'Oracle comme source pour AWS DMS

Les limites suivantes s'appliquent lors de l'utilisation d'une base de données Oracle comme source pour AWS DMS:

- AWS DMS prend en charge les types de données Oracle Extended dans les AWS DMS versions 3.5.0 et supérieures.
- AWS DMS ne prend pas en charge les noms d'objets longs (plus de 30 octets).
- AWS DMS ne prend pas en charge les index basés sur les fonctions.
- Si vous gérez la journalisation supplémentaire et que vous effectuez des transformations sur l'une des colonnes, assurez-vous que la journalisation supplémentaire est activée pour tous les champs et colonnes. Pour plus d'informations sur la configuration d'une journalisation supplémentaire, consultez les rubriques suivantes :
  - Pour une base de données source Oracle autogérée, consultez Configurez une journalisation supplémentaire.
  - Pour une base de données source Oracle AWS gérée, voir Configuration d'une source Oracle AWS gérée pour AWS DMS.
- AWS DMS ne prend pas en charge la base de données racine de conteneurs à locataires multiples (CDB$ROOT). Il prend en charge une PDB qui utilise Binary Reader.
- AWS DMS ne prend pas en charge les contraintes différées.
- Les LOB sécurisés sont pris en charge en mode LOB complet uniquement en effectuant une recherche d'objets LOB.
- AWS DMS prend en charge la `rename table` *`table-name`* `to` *`new-table-name`* syntaxe de toutes les versions Oracle 11 et supérieures prises en charge. Cette syntaxe n'est pas prise en charge pour les bases de données sources Oracle version 10.

- AWS DMS ne reproduit pas les résultats de l'instruction DDL. `ALTER TABLE ADD` *`column data_type`* `DEFAULT` *`default_value`* Au lieu de répliquer *`default_value`* vers la cible, il définit la nouvelle colonne sur `NULL`.
- Lorsque vous utilisez AWS DMS la version 3.4.7 ou supérieure, pour répliquer les modifications résultant d'opérations de partition ou de sous-partition, procédez comme suit avant de démarrer une tâche DMS.
  - Créez manuellement la structure de table partitionnée (DDL) ;
  - Assurez-vous que la DDL est la même sur la source Oracle et la cible Oracle ;
  - Définissez l'attribut de connexion supplémentaire `enableHomogenousPartitionOps=true`.

  Pour plus d'informations sur `enableHomogenousPartitionOps`, consultez Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS. Notez également que sur les tâches FULL+CDC, DMS ne réplique pas les modifications de données capturées dans le cadre des modifications mises en cache. Dans ce cas d'utilisation, recréez la structure de table sur la cible Oracle et rechargez les tables en question.

  Avant la AWS DMS version 3.4.7 :

  DMS ne réplique pas les modifications de données résultant d'opérations de partition ou de sous-partition (`ADD`, `DROP`, `EXCHANGE` et `TRUNCATE`). Ces mises à jour peuvent provoquer les erreurs suivantes pendant la réplication :

  - Pour les opérations `ADD`, les mises à jour et les suppressions des données ajoutées peuvent déclencher un avertissement « 0 lignes affectées ».
  - Pour les opérations `DROP` et `TRUNCATE`, les nouvelles insertions peuvent générer des erreurs de « doublons ».
  - Les opérations `EXCHANGE` peuvent déclencher à la fois un avertissement « 0 lignes affectées » et des erreurs « doublons ».

  Pour répliquer les modifications résultant d'opérations de partition ou de sous-partition, rechargez les tables en question. Après l'ajout d'une nouvelle partition vide, les opérations sur la partition récemment ajoutée sont répliquées vers la cible comme d'habitude.
- AWS DMS les versions antérieures à 3.4 ne prennent pas en charge les modifications de données sur la cible résultant de l'exécution de l'`CREATE TABLE AS`instruction sur la source. Toutefois, la nouvelle table est créée sur la cible.
- AWS DMS ne capture pas les modifications apportées par le `DBMS_REDEFINITION` package Oracle, par exemple les métadonnées de la table et le `OBJECT_ID` champ.

- AWS DMS mappe les colonnes BLOB et CLOB vides NULL sur la cible.
- Lors de la capture de modifications avec Oracle 11 LogMiner, une mise à jour sur une colonne CLOB dont la longueur de chaîne est supérieure à 1982 est perdue et la cible n'est pas mise à jour.
- Lors de la capture des données de modification (CDC), les mises à jour par lots des colonnes numériques définies comme clé primaire AWS DMS ne sont pas prises en charge.
- AWS DMS ne prend pas en charge certaines UPDATE commandes. L'exemple suivant est une commande UPDATE non prise en charge.

```
UPDATE TEST_TABLE SET KEY=KEY+1;
```

  Ici, TEST_TABLE est le nom de la table et KEY est une colonne numérique définie comme une clé primaire.
- AWS DMS ne prend pas en charge le mode LOB complet pour le chargement de colonnes LONG et LONG RAW. Vous pouvez plutôt utiliser le mode LOB limité pour migrer ces types de données vers une cible Oracle. En mode LOB limité, AWS DMS tronque à 64 Ko toutes les données que vous définissez comme des colonnes LONG ou LONG RAW de plus de 64 Ko.
- AWS DMS ne prend pas en charge le mode LOB complet pour le chargement de colonnes XMLTYPE. Vous pouvez plutôt utiliser le mode LOB limité pour migrer les colonnes XMLTYPE vers une cible Oracle. En mode LOB limité, DMS tronque toutes les données dont la taille est supérieure à la variable « Taille de LOB maximale » définie par l'utilisateur. La valeur maximale recommandée pour « Taille de LOB maximale » est de 100 Mo.
- AWS DMS ne reproduit pas les tables dont les noms contiennent des apostrophes.
- AWS DMS soutient le CDC à partir de vues matérialisées. Mais DMS ne prend pas en charge la CDC à partir d'une autre vue.
- AWS DMS ne prend pas en charge le CDC pour les tables organisées par index avec un segment de débordement.
- AWS DMS ne prend pas en charge l'Drop Partitionopération pour les tables partitionnées par référence avec la valeur enableHomogenousPartitionOps définie sur. true
- Lorsque vous utilisez Oracle LogMiner pour accéder aux journaux de journalisation, les limites suivantes AWS DMS s'appliquent :
  - Pour Oracle 12 uniquement, AWS DMS ne reproduit aucune modification apportée aux colonnes LOB.
  - Pour toutes les versions d'Oracle, AWS DMS ne reproduit pas le résultat des UPDATE opérations sur les colonnes LOB XMLTYPE et sur les colonnes.

  - AWS DMS ne prend pas en charge les transactions XA lors de la réplication lors de l'utilisation d'Oracle LogMiner.
  - Oracle LogMiner ne prend pas en charge les connexions à une base de données enfichable (PDB). Pour vous connecter à un PDB, accédez aux fichiers de journalisation à l'aide de Binary Reader.
  - Les opérations SHRINK SPACE ne sont pas prises en charge.
- Lorsque vous utilisez Binary Reader, il AWS DMS présente les limites suivantes :
  - Il ne prend pas en charge les clusters de tables.
  - Il ne prend en charge que les opérations `SHRINK SPACE` au niveau de la table. Ce niveau inclut la table complète, les partitions et les sous-partitions.
  - Il ne prend pas en charge les modifications apportées aux tables organisées par index avec compression de clé.
  - Il ne prend pas en charge l'implémentation de journaux redo en ligne sur les appareils bruts.
  - Binary Reader prend en charge le TDE uniquement pour les bases de données Oracle autogérées, car RDS for Oracle ne prend pas en charge la récupération des mots de passe de portefeuille pour les clés de chiffrement TDE.
- AWS DMS ne prend pas en charge les connexions à une source Oracle Amazon RDS à l'aide d'un proxy Oracle Automatic Storage Management (ASM).
- AWS DMS ne prend pas en charge les colonnes virtuelles.
- AWS DMS ne prend pas en charge le type de `ROWID` données ou les vues matérialisées basées sur une colonne ROWID.

  AWS DMS prend partiellement en charge Oracle Materialized Views. Pour les chargements complets, DMS peut effectuer une copie de chargement complet d'une vue matérialisée Oracle. DMS copie la vue matérialisée en tant que table de base sur le système cible et ignore les colonnes ROWID dans la vue matérialisée. Pour la réplication continue (CDC), DMS essaie de répliquer les modifications apportées aux données de la vue matérialisée, mais il est possible que les résultats ne soient pas idéaux. Plus précisément, si la vue matérialisée est complètement actualisée, DMS réplique les suppressions individuelles pour toutes les lignes, puis les insertions individuelles pour toutes les lignes. Cet exercice nécessite beaucoup de ressources et risque d'être peu performant pour les vues matérialisées comportant un grand nombre de lignes. Pour une réplication continue où les vues matérialisées sont actualisées rapidement, DMS essaie de traiter et de répliquer les modifications des données d'actualisation rapide. Dans les deux cas, DMS ignore toutes les colonnes ROWID dans la vue matérialisée.

- AWS DMS ne charge ni ne capture de tables temporaires globales.
- Pour les cibles S3 utilisant la réplication, activez la journalisation supplémentaire sur chaque colonne afin que les mises à jour des lignes source puissent capturer chaque valeur de colonne. Voici un exemple : `alter table yourtablename add supplemental log data (all) columns;`.
- La mise à jour d'une ligne comportant une clé unique composite contenant `null` ne peut pas être répliquée sur la cible.
- AWS DMS ne prend pas en charge l'utilisation de plusieurs clés de chiffrement Oracle TDE sur le même point de terminaison source. Chaque point de terminaison ne peut avoir qu'un seul attribut pour le nom de clé de chiffrement TDE « `securityDbEncryptionName` » et un seul mot de passe TDE pour cette clé.
- Lors de la réplication depuis Amazon RDS for Oracle, le TDE n'est pris en charge qu'avec un tablespace chiffré et avec Oracle. LogMiner
- AWS DMS ne prend pas en charge plusieurs opérations de renommage de table en succession rapide.
- Lorsque vous utilisez Oracle 19.0 comme source, les fonctionnalités suivantes AWS DMS ne sont pas prises en charge :
  - Redirection DML dans Data Guard
  - Tables hybrides partitionnées
  - Comptes Oracle de schéma uniquement
- AWS DMS ne prend pas en charge la migration de tables ou de vues de type `BIN$` ou`DR$`.
- À partir d'Oracle 18.x, la capture des données de modification (CDC) AWS DMS n'est pas prise en charge à partir d'Oracle Express Edition (Oracle Database XE).
- Lors de la migration de données à partir d'une colonne CHAR, DMS tronque les espaces de fin.
- AWS DMS ne prend pas en charge la réplication à partir de conteneurs d'applications.
- AWS DMS ne prend pas en charge l'exécution de la base de données Oracle Flashback et des points de restauration, car ces opérations affectent la cohérence des fichiers Oracle Redo Log.
- La procédure `INSERT` de chargement direct avec l'option d'exécution parallèle n'est pas prise en charge dans les cas suivants :
  - Tables non compressées de plus de 255 colonnes
  - Taille de ligne supérieure à 8 Ko
  - Tables HCC Exadata
  - Base de données exécutée sur la plateforme Big Endian

- Une table source n'ayant ni clé primaire ni clé unique nécessite l'activation de la journalisation supplémentaire ALL COLUMN. Des activités de journal redo supplémentaires seront créées et la latence de la CDC DMS peut augmenter.
- AWS DMS ne migre pas les données des colonnes invisibles de votre base de données source. Pour inclure ces colonnes dans la portée de migration, utilisez l'instruction `ALTER TABLE` pour les rendre visibles.

Prise en charge de SSL pour un point de terminaison Oracle

AWS DMS Les points de terminaison Oracle prennent en charge le protocole SSL V3 pour les modes `none` et `verify-ca` SSL. Pour utiliser SSL avec un point de terminaison Oracle, vous devez charger le portefeuille Oracle pour le point de terminaison au lieu des fichiers de certificat .pem.

Rubriques

- Utilisation d'un certificat existant pour Oracle SSL
- Utilisation d'un certificat auto-signé pour Oracle SSL

Utilisation d'un certificat existant pour Oracle SSL

Pour utiliser une installation de client Oracle existante afin de créer le fichier de portefeuille Oracle à partir du fichier de certificat CA, procédez comme suit.

Pour utiliser une installation de client Oracle existante pour Oracle SSL avec AWS DMS

1. Définissez la variable système `ORACLE_HOME` sur l'emplacement de votre répertoire `dbhome_1` en exécutant la commande suivante.

```
prompt>export ORACLE_HOME=/home/user/app/user/product/12.1.0/dbhome_1
```

2. Ajouter `$ORACLE_HOME/lib` à la variable système `LD_LIBRARY_PATH`.

```
prompt>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

3. Créez un répertoire pour le portefeuille Oracle dans `$ORACLE_HOME/ssl_wallet`.

```
prompt>mkdir $ORACLE_HOME/ssl_wallet
```

4. Placez le fichier `.pem` du certificat de l'autorité de certification dans le répertoire `ssl_wallet`. Si vous utilisez Amazon RDS, vous pouvez télécharger le fichier de certificat d'autorité de certification racine `rds-ca-2015-root.pem` hébergé par Amazon RDS. Pour plus d'informations sur le téléchargement de ce fichier, consultez Utilisation de SSL/TLS pour chiffrer une connexion à une instance de base de données dans le Guide de l'utilisateur Amazon RDS.
5. Exécutez les commandes suivantes pour créer le portefeuille Oracle.

```
prompt>orapki wallet create -wallet $ORACLE_HOME/ssl_wallet -auto_login_only
prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
   $ORACLE_HOME/ssl_wallet/ca-cert.pem -auto_login_only
```

Une fois que vous avez terminé les étapes précédentes, vous pouvez importer le fichier de portefeuille avec l'appel d'API `ImportCertificate` en spécifiant le paramètre de portefeuille de certificat. Vous pouvez ensuite utiliser le certificat de portefeuille importé lorsque vous sélectionnez `verify-ca` comme mode SSL lors de la création ou de la modification de votre point de terminaison Oracle.

> **Note**
>
> Les portefeuilles Oracle sont des fichiers binaires. AWS DMS accepte ces fichiers tels quels.

Utilisation d'un certificat auto-signé pour Oracle SSL

Pour utiliser un certificat auto-signé pour Oracle SSL, procédez comme suit en supposant que le mot de passe du portefeuille Oracle est `oracle123`.

Pour utiliser un certificat auto-signé pour Oracle SSL avec AWS DMS

1. Créez un répertoire que vous utiliserez pour travailler avec le certificat auto-signé.

```
mkdir -p /u01/app/oracle/self_signed_cert
```

2. Accédez au répertoire que vous avez créé à l'étape précédente.

```
cd /u01/app/oracle/self_signed_cert
```

3. Créez une clé racine.

```
openssl genrsa -out self-rootCA.key 2048
```

4. Auto-signez un certificat racine à l'aide de la clé racine que vous avez créée à l'étape précédente.

```
openssl req -x509 -new -nodes -key self-rootCA.key
        -sha256 -days 3650 -out self-rootCA.pem
```

Utilisez les paramètres d'entrée suivants.

- `Country Name (2 letter code) [XX]`, par exemple : AU
- `State or Province Name (full name) []`, par exemple : NSW
- `Locality Name (e.g., city) [Default City]`, par exemple : Sydney
- `Organization Name (e.g., company) [Default Company Ltd]`, par exemple : AmazonWebService
- `Organizational Unit Name (e.g., section) []`, par exemple : DBeng
- `Common Name (e.g., your name or your server's hostname) []`, par exemple : aws
- `Email Address []`, par exemple : abcd.efgh@amazonwebservice.com

5. Créez un répertoire de portefeuille Oracle pour la base de données Oracle.

```
mkdir -p /u01/app/oracle/wallet
```

6. Créez un nouveau portefeuille Oracle.

```
orapki wallet create -wallet "/u01/app/oracle/wallet" -pwd oracle123 -
auto_login_local
```

7. Ajoutez le certificat racine au portefeuille Oracle.

```
orapki wallet add -wallet "/u01/app/oracle/wallet" -pwd oracle123 -trusted_cert
-cert /u01/app/oracle/self_signed_cert/self-rootCA.pem
```

8. Affichez le contenu du portefeuille Oracle. La liste doit inclure le certificat racine.

```
orapki wallet display -wallet /u01/app/oracle/wallet -pwd oracle123
```

Par exemple, ce paramètre peut s'afficher comme suit.

```
Requested Certificates:
User Certificates:
Trusted Certificates:
Subject:        CN=aws,OU=DBeng,O= AmazonWebService,L=Sydney,ST=NSW,C=AU
```

9. Générez la demande de signature de certificat (CSR) à l'aide de l'utilitaire ORAPKI.

```
orapki wallet add -wallet "/u01/app/oracle/wallet" -pwd oracle123
-dn "CN=aws" -keysize 2048 -sign_alg sha256
```

10. Exécutez la commande suivante.

```
openssl pkcs12 -in /u01/app/oracle/wallet/ewallet.p12 -nodes -out /u01/app/oracle/
wallet/nonoracle_wallet.pem
```

Ce paramètre a un résultat similaire à ce qui suit.

```
Enter Import Password:
MAC verified OK
Warning unsupported bag type: secretBag
```

11. Mettez « dms » comme nom commun.

```
openssl req -new -key /u01/app/oracle/wallet/nonoracle_wallet.pem -out certdms.csr
```

Utilisez les paramètres d'entrée suivants.

- `Country Name (2 letter code) [XX]`, par exemple : `AU`
- `State or Province Name (full name) []`, par exemple : `NSW`
- `Locality Name (e.g., city) [Default City]`, par exemple : `Sydney`
- `Organization Name (e.g., company) [Default Company Ltd]`, par exemple : `AmazonWebService`
- `Organizational Unit Name (e.g., section) []`, par exemple : `aws`

- `Common Name (e.g., your name or your server's hostname) []`, par exemple : `aws`
- `Email Address []`, par exemple : abcd.efgh@amazonwebservice.com

Assurez-vous qu'il est différent de l'étape 4. Pour ce faire, remplacez par exemple le nom de l'unité organisationnelle par un autre nom, comme indiqué.

Entrez les attributs supplémentaires suivants à envoyer avec votre demande de certificat.

- `A challenge password []`, par exemple : `oracle123`
- `An optional company name []`, par exemple : `aws`

12. Obtenez la signature de certificat.

```
openssl req -noout -text -in certdms.csr | grep -i signature
```

La clé de signature de ce message est `sha256WithRSAEncryption`.

13. Exécutez la commande suivante pour générer le fichier de certificat (`.crt`).

```
openssl x509 -req -in certdms.csr -CA self-rootCA.pem -CAkey self-rootCA.key
-CAcreateserial -out certdms.crt -days 365 -sha256
```

Ce paramètre a un résultat similaire à ce qui suit.

```
Signature ok
subject=/C=AU/ST=NSW/L=Sydney/O=awsweb/OU=DBeng/CN=aws
Getting CA Private Key
```

14. Ajoutez le certificat au portefeuille.

```
orapki wallet add -wallet /u01/app/oracle/wallet -pwd oracle123 -user_cert -cert
 certdms.crt
```

15. Consultez le portefeuille. Il doit comporter deux entrées. Consultez le code suivant.

```
orapki wallet display -wallet /u01/app/oracle/wallet -pwd oracle123
```

16. Configurez le fichier `sqlnet.ora` (`$ORACLE_HOME/network/admin/sqlnet.ora`).

```
WALLET_LOCATION =
   (SOURCE =
     (METHOD = FILE)
     (METHOD_DATA =
       (DIRECTORY = /u01/app/oracle/wallet/)
     )
   )

SQLNET.AUTHENTICATION_SERVICES = (NONE)
SSL_VERSION = 1.0
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_CIPHER_SUITES = (SSL_RSA_WITH_AES_256_CBC_SHA)
```

17. Arrêtez l'écouteur Oracle.

```
lsnrctl stop
```

18. Ajoutez des entrées pour SSL dans le fichier `listener.ora` (`$ORACLE_HOME/network/admin/listener.ora`).

```
SSL_CLIENT_AUTHENTICATION = FALSE
WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /u01/app/oracle/wallet/)
    )
  )

SID_LIST_LISTENER =
 (SID_LIST =
  (SID_DESC =
   (GLOBAL_DBNAME = SID)
   (ORACLE_HOME = ORACLE_HOME)
   (SID_NAME = SID)
  )
 )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost.localdomain)(PORT = 1521))
```

```
      (ADDRESS = (PROTOCOL = TCPS)(HOST = localhost.localdomain)(PORT = 1522))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )
```

19. Configurez le fichier `tnsnames.ora` (`$ORACLE_HOME/network/admin/tnsnames.ora`).

```
<SID>=
(DESCRIPTION=
        (ADDRESS_LIST =
                (ADDRESS=(PROTOCOL = TCP)(HOST = localhost.localdomain)(PORT =
 1521))
        )
        (CONNECT_DATA =
                (SERVER = DEDICATED)
                (SERVICE_NAME = <SID>)
        )
)

<SID>_ssl=
(DESCRIPTION=
        (ADDRESS_LIST =
                (ADDRESS=(PROTOCOL = TCPS)(HOST = localhost.localdomain)(PORT =
 1522))
        )
        (CONNECT_DATA =
                (SERVER = DEDICATED)
                (SERVICE_NAME = <SID>)
        )
)
```

20. Redémarrez l'écouteur Oracle.

```
lsnrctl start
```

21. Affichez le statut de l'écouteur Oracle.

```
lsnrctl status
```

22. Testez la connexion SSL à la base de données à partir de localhost à l'aide de sqlplus et de l'entrée SSL tnsnames.

```
sqlplus -L ORACLE_USER@SID_ssl
```

23. Vérifiez que la connexion a été établie à l'aide de SSL.

```
SELECT SYS_CONTEXT('USERENV', 'network_protocol') FROM DUAL;

SYS_CONTEXT('USERENV','NETWORK_PROTOCOL')
----------------------------------------------------------------------------
tcps
```

24. Accédez au répertoire contenant le certificat auto-signé.

```
cd /u01/app/oracle/self_signed_cert
```

25. Créez un nouveau portefeuille client Oracle AWS DMS à utiliser.

```
orapki wallet create -wallet ./ -auto_login_only
```

26. Ajoutez le certificat racine auto-signé au portefeuille Oracle.

```
orapki wallet add -wallet ./ -trusted_cert -cert self-rootCA.pem -auto_login_only
```

27. Répertoriez le contenu du portefeuille Oracle AWS DMS à utiliser. La liste doit inclure le certificat racine auto-signé.

```
orapki wallet display -wallet ./
```

Ce paramètre a un résultat similaire à ce qui suit.

```
Trusted Certificates:
Subject:        CN=aws,OU=DBeng,O=AmazonWebService,L=Sydney,ST=NSW,C=AU
```

28. Téléchargez le portefeuille Oracle que vous venez de créer sur AWS DMS.

Méthodes de chiffrement prises en charge pour l'utilisation d'Oracle comme source pour AWS DMS

Dans le tableau suivant, vous trouverez les méthodes de chiffrement transparent des données (TDE) prises AWS DMS en charge lors de l'utilisation d'une base de données source Oracle.

Méthode d'accès aux journaux de journalisation	Espace disque logique TDE	Colonne TDE
Oracle LogMiner	Oui	Oui
Binary Reader	Oui	Oui

AWS DMS prend en charge Oracle TDE lorsque vous utilisez Binary Reader, à la fois au niveau des colonnes et au niveau de l'espace disque logique. Pour utiliser le chiffrement TDE avec AWS DMS, identifiez d'abord l'emplacement du portefeuille Oracle où sont stockés la clé de chiffrement TDE et le mot de passe TDE. Identifiez ensuite la clé de chiffrement TDE et le mot de passe appropriés pour le point de terminaison source Oracle.

Pour identifier et spécifier la clé de chiffrement et le mot de passe pour le chiffrement TDE

1. Exécutez la requête suivante pour trouver le portefeuille de chiffrement Oracle sur l'hôte de base de données Oracle.

   ```
   SQL> SELECT WRL_PARAMETER FROM V$ENCRYPTION_WALLET;

   WRL_PARAMETER
   --------------------------------------------------------------------------------
   /u01/oracle/product/12.2.0/dbhome_1/data/wallet/
   ```

   Dans cet exemple, l'emplacement du portefeuille est `/u01/oracle/product/12.2.0/dbhome_1/data/wallet/`.

2. Obtenez l'ID de la clé principale à l'aide de l'une des options de chiffrement suivantes, selon celle qui renvoie cette valeur.

   a. Pour le chiffrement au niveau de la table ou de la colonne, exécutez les requêtes suivantes.

   ```
   SQL> SELECT OBJECT_ID FROM ALL_OBJECTS
   WHERE OWNER='DMS_USER' AND OBJECT_NAME='TEST_TDE_COLUMN' AND
    OBJECT_TYPE='TABLE';

   OBJECT_ID
   ---------------
   81046
   SQL> SELECT MKEYID FROM SYS.ENC$ WHERE OBJ#=81046;
   ```

```
MKEYID
------------
AWGDC9glSk8Xv+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Dans cet exemple, l'ID de clé principale est `AWGDC9glSk8Xv+3bVveiVSg` (`MKEYID`). Si vous obtenez une valeur pour `MKEYID`, vous pouvez passer à l'étape 3. Sinon, poursuivez avec l'étape 2.2.

> **Note**
>
> Les caractères `'A'` de la chaîne de fin (`AAA...`) ne font pas partie de la valeur.

b. Pour le chiffrement au niveau de l'espace de table, exécutez les requêtes suivantes.

```
SQL> SELECT TABLESPACE_NAME, ENCRYPTED FROM dba_tablespaces;
TABLESPACE_NAME                ENC
------------------------------ ---
SYSTEM                         NO
SYSAUX                         NO
UNDOTBS1                       NO
TEMP                           NO
USERS                          NO
TEST_ENCRYT                    YES
SQL> SELECT name,utl_raw.cast_to_varchar2( utl_encode.base64_encode('01'||
substr(mkeyid,1,4))) ||
 utl_raw.cast_to_varchar2( utl_encode.base64_encode(substr(mkeyid,5,length(mkeyid))))
 masterkeyid_base64
FROM (SELECT t.name, RAWTOHEX(x.mkid) mkeyid FROM v$tablespace t, x$kcbtek x
 WHERE t.ts#=x.ts#)
WHERE name = 'TEST_ENCRYT';

NAME                           MASTERKEYID_BASE64
------------------------------ --------------------------------
TEST_ENCRYT                    AWGDC9glSk8Xv+3bVveiVSg=
```

Dans cet exemple, l'ID de clé principale est `AWGDC9glSk8Xv+3bVveiVSg` (`TEST_ENCRYT`). Si les étapes 2.1 et 2.2 renvoient une valeur, elles sont toujours identiques.

Le caractère `'='` de fin ne fait pas partie de la valeur.

3. À partir de la ligne de commande, répertoriez les entrées du portefeuille de chiffrement sur l'hôte de base de données Oracle source.

```
$ mkstore -wrl /u01/oracle/product/12.2.0/dbhome_1/data/wallet/ -list
Oracle Secret Store entries:
ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9glSk8Xv+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.DB.ENCRYPTION.AY1mRA8OXU9Qvzo3idU4OH4AAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.DB.ENCRYPTION.MASTERKEY
ORACLE.SECURITY.ID.ENCRYPTION.
ORACLE.SECURITY.KB.ENCRYPTION.
ORACLE.SECURITY.KM.ENCRYPTION.AY1mRA8OXU9Qvzo3idU4OH4AAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

   Recherchez l'entrée contenant l'ID de clé principale que vous avez trouvé à l'étape 2 (`AWGDC9glSk8Xv+3bVveiVSg`). Cette entrée est le nom de la clé de chiffrement TDE.

4. Consultez les détails de l'entrée que vous avez trouvés à l'étape précédente.

```
$ mkstore -wrl /u01/oracle/product/12.2.0/dbhome_1/data/wallet/ -viewEntry
 ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9glSk8Xv+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Oracle Secret Store Tool : Version 12.2.0.1.0
Copyright (c) 2004, 2016, Oracle and/or its affiliates. All rights reserved.
Enter wallet password:
ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9glSk8Xv+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 = AEMAASAASGYs0phWHfNt9J5mEMkkegGFiD4LLfQszDojgDzbfoYDEACv0x3pJC+UGD/
PdtE2jLIcBQcAeHgJChQGLA==
```

   Entrez le mot de passe du portefeuille pour voir le résultat.

   Ici, la valeur à droite de `'='` est le mot de passe TDE.

5. Spécifiez le nom de la clé de chiffrement TDE pour le point de terminaison source Oracle en définissant l'attribut de connexion supplémentaire `securityDbEncryptionName`.

```
securityDbEncryptionName=ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9glSk8Xv
+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

6. Indiquez le mot de passe TDE associé à cette clé sur la console dans le champ Mot de passe de la source Oracle. Utilisez l'ordre suivant pour mettre en forme les valeurs de mot de passe séparées par des virgules, qui se terminent par la valeur du mot de passe TDE.

```
Oracle_db_password,ASM_Password,AEMAASAASGYs0phWHfNt9J5mEMkkegGFiD4LLfQszDojgDzbfoYDEACv0x3
+UGD/PdtE2jLIcBQcAeHgJChQGLA==
```

Spécifiez les valeurs de mot de passe dans cet ordre quelle que soit la configuration de votre base de données Oracle. Par exemple, si vous utilisez TDE, mais que la base de données Oracle n'utilise pas ASM, spécifiez les valeurs de mot de passe dans l'ordre suivant séparé par des virgules.

```
Oracle_db_password,,AEMAASAASGYs0phWHfNt9J5mEMkkegGFiD4LLfQszDojgDzbfoYDEACv0x3pJC
+UGD/PdtE2jLIcBQcAeHgJChQGLA==
```

Si les informations d'identification TDE que vous spécifiez sont incorrectes, la tâche de AWS DMS migration n'échoue pas. Toutefois, la tâche ne lit pas et n'applique pas non plus les modifications de réplication continue à la base de données cible. Après avoir démarré la tâche, surveillez les Statistiques de table sur la page de la tâche de migration de la console pour vous assurer que les modifications sont répliquées.

Si un administrateur de base de données modifie les valeurs d'informations d'identification TDE pour la base de données Oracle pendant l'exécution de la tâche, celle-ci échoue. Le message d'erreur contient le nouveau nom de clé de chiffrement TDE. Pour spécifier de nouvelles valeurs et redémarrer la tâche, procédez comme suit.

Important

Vous ne pouvez pas manipuler un portefeuille TDE créé dans un emplacement Oracle Automatic Storage Management (ASM), car les commandes au niveau du système d'exploitation telles que `cp`, `mv`, `orapki` et `mkstore` corrompent les fichiers du portefeuille stockés dans un emplacement ASM. Cette restriction est spécifique aux fichiers de portefeuille TDE stockés dans un emplacement ASM uniquement, mais pas aux fichiers de portefeuille TDE stockés dans un répertoire local du système d'exploitation.
Pour manipuler un portefeuille TDE stocké dans ASM à l'aide de commandes au niveau du système d'exploitation, créez un magasin de clés local et fusionnez-le avec le magasin de clés ASM, comme suit :

1. Créez un magasin de clés local.

```
ADMINISTER KEY MANAGEMENT create keystore file system wallet location
 identified by wallet password;
```

2. Fusionnez le magasin de clés ASM avec le magasin de clés local.

```
ADMINISTER KEY MANAGEMENT merge keystore ASM wallet location identified
 by wallet password into existing keystore file system wallet location
 identified by wallet password with backup;
```

Ensuite, pour répertorier les entrées du portefeuille de chiffrement et le mot de passe TDE, exécutez les étapes 3 et 4 sur le magasin de clés local.

Méthodes de compression prises en charge pour l'utilisation d'Oracle comme source pour AWS DMS

Dans le tableau suivant, vous trouverez les méthodes de compression prises AWS DMS en charge lors de l'utilisation d'une base de données source Oracle. Comme le montre le tableau, la prise en charge de la compression dépend à la fois de la version de votre base de données Oracle et de la configuration de DMS pour utiliser Oracle LogMiner pour accéder aux journaux de journalisation.

Version	Base	OLTP	HCC (à partir d'Oracle 11g R2 ou ultérieur)	Autres
Oracle 10	Non	N/A	N/A	Non
Oracle 11 ou version ultérieure — Oracle LogMiner	Oui	Oui	Oui	Oui — Toutes les méthodes de compression prises en charge par Oracle LogMiner.

Version	Base	OLTP	HCC (à partir d'Oracle 11g R2 ou ultérieur)	Autres
Oracle 11 ou version ultérieure : Binary Reader	Oui	Oui	Oui : pour plus d'informations, consultez la remarque suivante.	Oui

> **Note**
>
> Lorsque le point de terminaison source Oracle est configuré pour utiliser Binary Reader, le niveau Query Low de la méthode de compression HCC est pris en charge uniquement pour les tâches à pleine charge.

Réplication de tables imbriquées en utilisant Oracle comme source pour AWS DMS

AWS DMS prend en charge la réplication des tables Oracle contenant des colonnes qui sont des tables imbriquées ou des types définis. Pour activer cette fonctionnalité, ajoutez le paramètre d'attribut de connexion supplémentaire suivant au point de terminaison source Oracle.

```
allowSelectNestedTables=true;
```

AWS DMS crée les tables cibles à partir des tables imbriquées Oracle en tant que tables parent et enfant normales sur la cible sans contrainte unique. Pour accéder aux données correctes sur la cible, joignez les tables parent et enfant. Pour ce faire, créez d'abord manuellement un index non unique sur la colonne `NESTED_TABLE_ID` de la table enfant cible. Vous pouvez ensuite utiliser la colonne `NESTED_TABLE_ID` dans la clause de jointure `ON` avec la colonne parent qui correspond au nom de la table enfant. En outre, la création d'un tel index améliore les performances lorsque les données de la table enfant cible sont mises à jour ou supprimées par AWS DMS. Pour obtenir un exemple, consultez Exemple de jointure pour les tables parent et enfant sur la cible.

Nous vous recommandons de configurer la tâche pour qu'elle s'arrête une fois la charge complète terminée. Ensuite, créez ces index non uniques pour toutes les tables enfants répliquées sur la cible et reprenez la tâche.

Si une table imbriquée capturée est ajoutée à une table parent existante (capturée ou non capturée), elle AWS DMS est gérée correctement. Toutefois, l'index non unique de la table cible correspondante n'est pas créé. Dans ce cas, si la table enfant cible devient extrêmement volumineuse, les performances peuvent être affectées. Dans ce cas, nous vous recommandons d'arrêter la tâche, de créer l'index, puis de reprendre la tâche.

Une fois les tables imbriquées répliquées vers la cible, faites en sorte que l'administrateur de base de données exécute une jointure sur les tables parent et enfant correspondantes pour aplatir les données.

Conditions préalables à la réplication des tables imbriquées Oracle en tant que source

Assurez-vous de répliquer les tables parentes pour toutes les tables imbriquées répliquées. Incluez à la fois les tables parents (les tables contenant la colonne de table imbriquée) et les tables enfants (c'est-à-dire imbriquées) dans les AWS DMS mappages de tables.

Types de tables imbriquées Oracle pris en charge en tant que source

AWS DMS prend en charge les types de tables imbriquées Oracle suivants en tant que source :

- Type de données
- Objet défini par l'utilisateur

Limitations de la prise en charge par AWS DMS des tables imbriquées Oracle en tant que source

AWS DMS présente les limites suivantes en ce qui concerne sa prise en charge des tables imbriquées Oracle en tant que source :

- AWS DMS ne prend en charge qu'un seul niveau d'imbrication de tables.
- AWS DMS le mappage des tables ne vérifie pas que la ou les tables parent et enfant sont sélectionnées pour la réplication. Autrement dit, il est possible de sélectionner une table parent sans enfant ou une table enfant sans parent.

Comment AWS DMS réplique les tables imbriquées Oracle en tant que source

AWS DMS réplique les tables parentes et imbriquées vers la cible comme suit :

- AWS DMS crée la table parent identique à la source. Il définit ensuite la colonne imbriquée dans le parent comme `RAW(16)` et inclut une référence aux tables imbriquées du parent dans sa colonne `NESTED_TABLE_ID`.

- AWS DMS crée la table enfant identique à la source imbriquée, mais avec une colonne supplémentaire nomméeNESTED_TABLE_ID. Cette colonne a le même type et la même valeur que la colonne imbriquée parent correspondante et a la même signification.

Exemple de jointure pour les tables parent et enfant sur la cible

Pour aplatir la table parent, exécutez une jointure entre les tables parent et enfant, comme illustré dans l'exemple suivant :

1. Créez la table Type.

```
CREATE OR REPLACE TYPE NESTED_TEST_T AS TABLE OF VARCHAR(50);
```

2. Créez la table parent avec une colonne de type NESTED_TEST_T, tel que défini précédemment.

```
CREATE TABLE NESTED_PARENT_TEST (ID NUMBER(10,0) PRIMARY KEY, NAME NESTED_TEST_T)
 NESTED TABLE NAME STORE AS NAME_KEY;
```

3. Aplatissez la table NESTED_PARENT_TEST à l'aide d'une jointure avec la table NAME_KEY enfant où CHILD.NESTED_TABLE_ID correspond à PARENT.NAME.

```
SELECT … FROM NESTED_PARENT_TEST PARENT, NAME_KEY CHILD WHERE CHILD.NESTED_
TABLE_ID = PARENT.NAME;
```

Stockage de REDO sur Oracle ASM lors de l'utilisation d'Oracle comme source pour AWS DMS

Pour les sources Oracle à forte génération de REDO, le stockage de REDO sur Oracle ASM peut améliorer les performances, en particulier dans une configuration RAC, car vous pouvez configurer DMS pour distribuer les lectures ASM REDO sur tous les nœuds ASM.

Pour utiliser cette configuration, utilisez l'attribut de connexion asmServer. Par exemple, la chaîne de connexion suivante distribue les lectures DMS REDO sur 3 nœuds ASM :

```
asmServer=(DESCRIPTION=(CONNECT_TIMEOUT=8)(ENABLE=BROKEN)(LOAD_BALANCE=ON)(FAILOVER=ON)
(ADDRESS_LIST=
(ADDRESS=(PROTOCOL=tcp)(HOST=asm_node1_ip_address)(PORT=asm_node1_port_number))
(ADDRESS=(PROTOCOL=tcp)(HOST=asm_node2_ip_address)(PORT=asm_node2_port_number))
(ADDRESS=(PROTOCOL=tcp)(HOST=asm_node3_ip_address)(PORT=asm_node3_port_number)))
```

```
(CONNECT_DATA=(SERVICE_NAME=+ASM)))
```

Lorsque vous utilisez NFS pour stocker Oracle REDO, il est important de vous assurer que les correctifs client DNFS (Direct NFS) appropriés sont appliqués, en particulier les correctifs qui corrigent le bogue Oracle 25224242. Pour plus d'informations, consultez la publication Oracle suivante concernant les correctifs liés au client Direct NFS, intitulée Recommended Patches for Direct NFS Client.

En outre, pour améliorer les performances de lecture NFS, nous vous recommandons d'augmenter la valeur de `rsize` et de `wsize` dans `fstab` pour le volume NFS, comme indiqué dans l'exemple suivant.

```
NAS_name_here:/ora_DATA1_archive /u09/oradata/DATA1 nfs
 rw,bg,hard,nointr,tcp,nfsvers=3,_netdev,
timeo=600,rsize=262144,wsize=262144
```

Ajustez également la valeur de `tcp-max-xfer-size` comme suit :

```
vserver nfs modify -vserver vserver -tcp-max-xfer-size 262144
```

Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données source Oracle comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison source à l'aide de la AWS DMS console ou à l'aide de la `create-endpoint` commande dans le AWS CLI, avec la syntaxe `--oracle-settings '{"EndpointSetting": "value", ...}'` JSON.

Les paramètres de point de terminaison que vous pouvez utiliser avec Oracle en tant que source sont indiqués dans le tableau suivant.

Name (Nom)	Description
`AccessAlternateDirectly`	Définissez cet attribut sur false pour utiliser Binary Reader afin de capturer les données de modification pour Amazon RDS for Oracle en tant que source. Cette opération indique à l'instance DMS qu'elle ne doit pas accéder aux journaux redo via un remplacement de préfixe

Name (Nom)	Description
	de chemin spécifié en utilisant un accès direct aux fichiers. Pour plus d'informations, consultez Configuration d'une tâche CDC pour utiliser Binary Reader avec une source RDS pour Oracle pour AWS DMS. Valeur par défaut : true Valeurs valides : true/false Exemple : `--oracle-settings '{"AccessAlternateDirectly": false}'`
`AdditionalArchivedLogDestId`	Définissez cet attribut avec `ArchivedLogDestId` dans une configuration principale/de secours. Cet attribut est utile lors d'un basculement lorsqu'une base de données Oracle Data Guard est utilisée en tant que source. Dans ce cas, vous AWS DMS devez savoir de quelle destination obtenir les journaux de restauration des archives pour lire les modifications. En effet, après le basculement, l'instance précédemment principale est désormais une instance de secours. Bien qu'il AWS DMS supporte l'utilisation de l'`RESETLOGS` option Oracle pour ouvrir la base de données, ne l'utilisez jamais `RESETLOGS` sauf si cela est nécessaire. Pour plus d'informations sur `RESETLOGS`, consultez RMAN Data Repair Concepts dans le Guide de l'utilisateur Oracle® Database Backup and Recovery. Valeurs valides : ID de destination d'archive Exemple : `--oracle-settings '{"AdditionalArchivedLogDestId": 2}'`

Name (Nom)	Description
`AddSupplementalLogging`	Définissez cet attribut pour configurer la journalisation supplémentaire au niveau de la table pour la base de données Oracle. Cet attribut active l'une des options suivantes sur toutes les tables sélectionnées pour une tâche de migration, en fonction des métadonnées des tables : • Journalisation supplémentaire des COLONNES PRIMARY KEY • Journalisation supplémentaire des COLONNES UNIQUE KEY • Journalisation supplémentaire de TOUTES LES COLONNES Valeur par défaut : false Valeurs valides : true/false Exemple : `--oracle-settings '{"AddSupplementalLogging": false}'` **Note** Si vous utilisez cette option, vous devez encore activer la journalisation supplémentaire au niveau de la base de données, comme indiqué précédemment.
`AllowSelectNestedTables`	Définissez cet attribut sur true pour activer la réplication des tables Oracle contenant des colonnes imbriquées ou des types définis. Pour plus d'informations, consultez Réplication de tables imbriquées en utilisant Oracle comme source pour AWS DMS. Valeur par défaut : false Valeurs valides : true/false Exemple : `--oracle-settings '{"AllowSelectNestedTables": true}'`

Name (Nom)	Description
`ArchivedLogDestId`	Spécifie l'ID de la destination des journaux redo archivés. Cette valeur doit être identique à un des nombres de la colonne dest_id de la vue v $archived_log. Si vous travaillez avec une destination de journal redo supplémentaire, nous vous recommandons d'utiliser l'attribut `AdditionalArchivedLogDestId` pour spécifier l'ID de destination supplémentaire. Cela améliore les performances en garantissant l'accès aux journaux appropriés dès le départ. Valeur par défaut : 1 Valeurs valides : nombre Exemple : `--oracle-settings '{"ArchivedLogDestId": 1}'`
`ArchivedLogsOnly`	Lorsque ce champ est défini sur Y, il accède AWS DMS uniquement aux journaux de journalisation archivés. Si les journaux de restauration archivés sont stockés uniquement sur Oracle ASM, le compte AWS DMS utilisateur doit disposer de privilèges ASM. Valeur par défaut : N Valeurs valides : Y/N Exemple : `--oracle-settings '{"ArchivedLogsOnly": Y}'`

Name (Nom)	Description
`asmUsePLSQLArray` (ECA uniquement)	Utilisez cet attribut de connexion supplémentaire (ECA) lors de la capture des modifications de source avec BinaryReader. Ce paramètre permet à DMS de mettre en mémoire tampon 50 lectures au niveau ASM par thread de lecture unique tout en contrôlant le nombre de threads à l'aide de l'attribut `parallelASMReadThread` . Lorsque vous définissez cet attribut, le lecteur AWS DMS binaire utilise un bloc PL/SQL anonyme pour capturer les données de restauration et les renvoyer à l'instance de réplication sous forme de grande mémoire tampon. Cela réduit le nombre d'allers retours jusqu'à la source. Cela peut améliorer de manière significative les performances de capture de source, mais cela entraîne une consommation de mémoire PGA plus élevée sur l'instance ASM. Des problèmes de stabilité peuvent survenir si la mémoire cible n'est pas suffisante. Vous pouvez utiliser la formule suivante pour estimer l'utilisation totale de la mémoire PGA sur l'instance ASM par tâche DMS : `number_of_redo_threads * parallelASMReadThreads * 7 MB` Valeur par défaut : false Valeurs valides : true/false Exemple d'ECA : `asmUsePLSQLArray=true;`
`ConvertTimestampWithZoneToUTC`	Définissez cet attribut sur `true` pour convertir la valeur d'horodatage des colonnes « TIMESTAMP WITH TIME ZONE » et « TIMESTAMP WITH LOCAL TIME ZONE » au format UTC. Par défaut, la valeur de cet attribut est « false » et les données seront répliquées en utilisant le fuseau horaire de la base de données source. Valeur par défaut : false Valeurs valides : true/false Exemple : `--oracle-settings '{"ConvertTimestampWithZoneToUTC": true}'`

Name (Nom)	Description
`EnableHomogenousPartitionOps`	Définissez cet attribut sur `true` pour permettre la réplication des opérations DDL de partition et de sous-partition Oracle pour une migration Oracle homogène. Notez que cette fonctionnalité et cette amélioration ont été introduites dans la AWS DMS version 3.4.7. Valeur par défaut : false Valeurs valides : true/false Exemple : `--oracle-settings '{"EnableHomogenousPartitionOps": true}'`
`EnableHomogenousTablespace`	Définissez cet attribut pour activer la réplication homogène d'un espace de table et créer des tables ou des index existants sous le même espace de table sur la cible. Valeur par défaut : false Valeurs valides : true/false Exemple : `--oracle-settings '{"EnableHomogenousTablespace": true}'`

Name (Nom)	Description
`EscapeCharacter`	Définissez cet attribut sur un caractère d'échappement. Ce caractère d'échappement vous permet de faire en sorte qu'un caractère générique unique se comporte comme un caractère normal dans les expressions de mappage de table. Pour plus d'informations, consultez Caractères génériques dans le mappage de table. Valeur par défaut : Null Valeurs valides : tout caractère autre qu'un caractère générique Exemple : `--oracle-settings '{"EscapeCharacter": "#"}'` **Note** Vous pouvez uniquement utiliser `escapeCharacter` pour les noms de table. Il n'exclut pas les caractères des noms de schéma ou de colonne.
`ExposeViews`	Utilisez cet attribut pour extraire les données d'une vue une fois ; vous ne pouvez pas l'utiliser pour la réplication continue. Lorsque vous extrayez les données d'une vue, la vue est représentée sous la forme d'une table sur le schéma cible. Valeur par défaut : false Valeurs valides : true/false Exemple : `--oracle-settings '{"ExposeViews": true}'`

Name (Nom)	Description
`ExtraArchivedLogDestIds`	Spécifie les ID d'une ou de plusieurs destinations pour un ou plusieurs journaux redo archivés. Ces ID sont les valeurs de la colonne dest_id dans la vue v$archived_log. Utilisez ce paramètre avec l'attribut de connexion ArchivedLogDestId supplémentaire dans une primary-to-single configuration ou une primary-to-multiple-standby configuration. Ce paramètre est utile lors d'un basculement lorsque vous utilisez une base de données Oracle Data Guard comme source. Dans ce cas, il AWS DMS a besoin d'informations sur la destination à partir de laquelle obtenir les journaux de restauration des archives pour lire les modifications. AWS DMS en a besoin car après le basculement, l'instance principale précédente est une instance de secours. Valeurs valides : ID de destination d'archive Exemple : `--oracle-settings '{"ExtraArchivedLogDestIds": 1}'`
`FailTasksOnLobTruncation`	Lorsque la valeur `true` est définie, cet attribut provoque l'échec d'une tâche si la taille réelle d'une colonne LOB est supérieure à la valeur `LobMaxSize` spécifiée. Si une tâche est définie sur le mode LOB limité et que cette option a pour valeur `true`, la tâche échoue au lieu de tronquer les données LOB. Valeur par défaut : false Valeurs valides : booléen Exemple : `--oracle-settings '{"FailTasksOnLobTruncation": true}'`

Name (Nom)	Description
`filterTransactionsOfUser` (ECA uniquement)	Utilisez cet attribut de connexion supplémentaire (ECA) pour permettre à DMS d'ignorer les transactions d'un utilisateur spécifié lors de la réplication des données d'Oracle lors de l'utilisation. LogMiner Vous pouvez transmettre des valeurs de nom d'utilisateur séparées par des virgules, mais elles doivent être entièrement en majuscules. Exemple d'ECA : `filterTransactionsOfUser=` *`USERNAME`*`;`
`NumberDataTypeScale`	Spécifie l'échelle de nombres. Vous pouvez sélectionner une échelle jusqu'à 38, -1 pour FLOAT ou -2 pour VARCHAR. Par défaut, le type de données NUMBER est converti en précision 38, échelle 10. Valeur par défaut : 10 Valeurs valides : -2 à 38 (-2 pour VARCHAR, -1 pour FLOAT) Exemple : `--oracle-settings '{"NumberDataTypeScale": 12}'` **Note** Sélectionnez une combinaison d'échelle de précision, -1 (FLOAT) ou -2 (VARCHAR). DMS prend en charge toutes les combinaisons d'échelle de précision prises en charge par Oracle. Si la précision est supérieure ou égale à 39, sélectionnez -2 (VARCHAR). Le NumberDataTypeScale paramètre de la base de données Oracle est utilisé uniquement pour le type de données NUMBER (sans précision ni définition d'échelle explicites).
`OpenTransactionWindow`	Fournit le délai en minutes pour vérifier l'existence de transactions ouvertes pour une tâche de CDC uniquement. Valeur par défaut : 0 Valeurs valides : entier compris entre 0 et 240 Exemple : `openTransactionWindow=15;`

Name (Nom)	Description
`OraclePathPrefix`	Définissez cet attribut de chaîne sur la valeur requise afin d'utiliser Binary Reader pour capturer les données modifiées pour un Amazon RDS for Oracle en tant que source. Cette valeur spécifie la racine Oracle utilisée par défaut pour accéder aux journaux redo. Pour plus d'informations, consultez Configuration d'une tâche CDC pour utiliser Binary Reader avec une source RDS pour Oracle pour AWS DMS. Valeur par défaut : none Valeur valide : /rdsdbdata/db/ORCL_A/ Exemple : `--oracle-settings '{"OraclePathPrefix": "  /rdsdbdata/db/ORCL_A/  "}'`
`ParallelASMReadThreads`	Définissez cet attribut pour modifier le nombre de threads configurés par DMS pour effectuer une capture des données de modification (CDC) à l'aide d'Oracle Automatic Storage Management (ASM). Vous pouvez spécifier un nombre entier compris entre 2 (par défaut) et 8 (maximum) . Utilisez cet attribut avec l'attribut `ReadAheadBlocks` . Pour plus d'informations, consultez Configuration d'une tâche CDC pour utiliser Binary Reader avec une source RDS pour Oracle pour AWS DMS. Valeur par défaut : 2 Valeurs valides : un nombre entier compris entre 2 à 8 Exemple : `--oracle-settings '{"ParallelASMReadThreads": 6;}'`

Name (Nom)	Description
`ReadAheadBlocks`	Définissez cet attribut pour modifier le nombre de blocs de lecture anticipée configurés par DMS pour effectuer une CDC à l'aide d'Oracle Automatic Storage Management (ASM) et d'un stockage NAS non ASM. Vous pouvez spécifier un nombre entier compris entre 1000 (valeur par défaut) et 200 000 (maximum). Utilisez cet attribut avec l'attribut `ParallelASMReadThreads` . Pour plus d'informations, consultez Configuration d'une tâche CDC pour utiliser Binary Reader avec une source RDS pour Oracle pour AWS DMS. Valeur par défaut : 1000 Valeurs valides : Un nombre entier compris entre 1000 à 200 000 Exemple : `--oracle-settings '{"ReadAheadBlocks": 150000}'`
`ReadTableSpaceName`	Lorsque la valeur `true` est définie, cet attribut prend en charge la réplication d'espace de table. Valeur par défaut : false Valeurs valides : booléen Exemple : `--oracle-settings '{"ReadTableSpaceName": true}'`

Name (Nom)	Description
`ReplacePathPrefix`	Définissez cet attribut sur true (vrai) pour utiliser Binary Reader afin de capturer les données modifiées pour un Amazon RDS for Oracle en tant que source. Ce paramètre indique à l'instance DMS qu'elle doit remplacer la racine Oracle par défaut par le paramètre `UsePathPrefix` pour accéder aux journaux redo. Pour plus d'informations, consultez Configuration d'une tâche CDC pour utiliser Binary Reader avec une source RDS pour Oracle pour AWS DMS. Valeur par défaut : false Valeurs valides : true/false Exemple : `--oracle-settings '{"ReplacePathPrefix": true}'`
`RetryInterval`	Spécifie le nombre de secondes que le système attend avant de renvoyer une requête. Valeur par défaut : 5 Valeurs valides : nombres à partir de 1 Exemple : `--oracle-settings '{"RetryInterval": 6}'`

Name (Nom)	Description
`SecurityDbEncryptionName`	Spécifie le nom d'une clé utilisée pour le chiffrement transparent des données (TDE) des colonnes et de l'espace disque logique dans la base de données source Oracle. Pour de plus amples informations sur la définition de cet attribut et de son mot de passe associé sur le point de terminaison source Oracle, veuillez consulter Méthodes de chiffrement prises en charge pour l'utilisation d'Oracle comme source pour AWS DMS. Valeur par défaut : "" Valeurs valides : string Exemple : `--oracle-settings '{"SecurityDbEncryptionName": "ORACLE.SECURITY.DB.ENCRYPTION.Adg8m2dhkU/0v/m5QUaaNJEAAAAAAAAAAAAAAAAAAAAAAAAAAA"}'`
`SpatialSdo2GeoJsonFunctionName`	Pour les sources Oracle version 12.1 ou antérieures migrant vers des cibles PostgreSQL, utilisez cet attribut pour convertir SDO_GEOMETRY au format GEOJSON. Par défaut, AWS DMS appelle la fonction `SDO2GEOJSON` personnalisée qui doit être présente et accessible à l' AWS DMS utilisateur. Vous pouvez également créer votre propre fonction personnalisée qui imite le fonctionnement de `SDOGEOJSON` et définir `SpatialSdo2GeoJsonFunctionName` pour l'appeler à la place. Valeur par défaut : SDO2GEOJSON Valeurs valides : string Exemple : `--oracle-settings '{"SpatialSdo2GeoJsonFunctionName": "myCustomSDO2GEOJSONFunction"}'`

Name (Nom)	Description
`StandbyDelayTime`	Utilisez cet attribut pour spécifier une durée en minutes pour le retard de synchronisation de l'instance de secours. Si la source est une base de données de secours Active Data Guard, utilisez cet attribut pour spécifier le décalage entre les bases de données principale et de secours. Dans AWS DMS, vous pouvez créer une tâche Oracle CDC qui utilise une instance de secours Active Data Guard comme source pour répliquer les modifications en cours. Cela vous permet de ne plus avoir besoin de vous connecter à une base de données active pouvant être en production. Valeur par défaut : 0 Valeurs valides : nombre Exemple : `--oracle-settings '{"StandbyDelayTime": 1}'` Remarque : lorsque vous utilisez DMS 3.4.6, 3.4.7 ou ultérieur, l'utilisation de ce paramètre de connexion est facultative. Dans les dernières versions de DMS 3.4.6 et 3.4.7, *`dms_user`* devrait avoir l'autorisation `select` sur `V_$DATAGUARD_STATS` , ce qui permet à DMS de calculer le délai de veille.
`UseAlternateFolderForOnline`	Définissez cet attribut sur true (vrai) pour utiliser Binary Reader afin de capturer les données modifiées pour un Amazon RDS for Oracle en tant que source. Cette opération indique à l'instance DMS qu'elle doit utiliser un remplacement de préfixe spécifié pour accéder à tous les journaux redo en ligne. Pour plus d'informations, consultez Configuration d'une tâche CDC pour utiliser Binary Reader avec une source RDS pour Oracle pour AWS DMS. Valeur par défaut : false Valeurs valides : true/false Exemple : `--oracle-settings '{"UseAlternateFolderForOnline": true}'`

Name (Nom)	Description
`UseBfile`	Définissez cet attribut sur Y pour capturer les données modifiées à l'aide de l'utilitaire Binary Reader. Attribuez la valeur N à `UseLogminerReader` pour définir cet attribut sur Y. Pour utiliser Binary Reader avec Amazon RDS for Oracle en tant que source, définissez des attributs supplémentaires. Pour plus d'informations sur ce paramètre et l'utilisation d'Oracle Automatic Storage Management (ASM), consultez Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC. Remarque : lorsque vous définissez cette valeur en tant qu'attribut de connexion supplémentaire (ECA), les valeurs valides sont « Y » et « N ». Lorsque vous définissez cette valeur en tant que paramètre de point de terminaison, les valeurs valides sont `true` et `false`. Valeur par défaut : N Valeurs valides : Y/N (lorsque vous définissez cette valeur en tant qu'ECA) ; true/false (lorsque vous définissez cette valeur en tant que paramètre de point de terminaison). Exemple : `--oracle-settings '{"UseBfile": Y}'`

Name (Nom)	Description
`UseLogminerReader`	Définissez cet attribut sur Y pour capturer les données de modification à l'aide de l' LogMiner utilitaire (valeur par défaut). Définissez cette option sur N si vous voulez qu' AWS DMS accède aux journaux redo en tant que fichier binaire. Lorsque vous définissez cette option sur N, ajoutez également le paramètre useBfile=Y. Pour plus d'informations sur ce paramètre et sur l'utilisation d'Oracle Automatic Storage Management (ASM), consultez Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC. Remarque : lorsque vous définissez cette valeur en tant qu'attribut de connexion supplémentaire (ECA), les valeurs valides sont « Y » et « N ». Lorsque vous définissez cette valeur en tant que paramètre de point de terminaison, les valeurs valides sont `true` et `false`. Valeur par défaut : Y Valeurs valides : Y/N (lorsque vous définissez cette valeur en tant qu'ECA) ; true/false (lorsque vous définissez cette valeur en tant que paramètre de point de terminaison). Exemple : `--oracle-settings '{"UseLogminerReader": Y}'`
`UsePathPrefix`	Définissez cet attribut de chaîne sur la valeur requise afin d'utiliser Binary Reader pour capturer les données modifiées pour un Amazon RDS for Oracle en tant que source. Cette valeur spécifie le préfixe de chemin utilisé pour remplacer la racine Oracle utilisée par défaut pour accéder aux journaux redo. Pour plus d'informations, consultez Configuration d'une tâche CDC pour utiliser Binary Reader avec une source RDS pour Oracle pour AWS DMS. Valeur par défaut : none Valeur valide : /rdsdbdata/log/ Exemple : `--oracle-settings '{"UsePathPrefix": "  /rdsdbdata/log/ "}'`

Types de données sources pour Oracle

Le point de terminaison Oracle pour AWS DMS prend en charge la plupart des types de données Oracle. Le tableau suivant indique les types de données source Oracle pris en charge lors de l'utilisation AWS DMS et le mappage par défaut aux types de AWS DMS données.

> **Note**
>
> À l'exception des types de données LONG et LONG RAW, lors de la réplication d'une source Oracle vers une cible Oracle (réplication homogène), tous les types de données source et cibles seront identiques. Toutefois, le type de données LONG sera mappé aux objets CLOB et le type de données LONG RAW sera mappé aux objets BLOB.

Pour en savoir plus sur la façon d'afficher le type de données qui est mappé dans la cible, consultez la section concernant le point de terminaison cible que vous utilisez.

Pour plus d'informations sur AWS DMS les types de données, consultezTypes de données pour AWS Database Migration Service.

Type de données Oracle	AWS DMS type de données
BINARY_FLOAT	REAL4
BINARY_DOUBLE	REAL8
BINAIRE	BYTES
FLOAT (P)	Si la précision est inférieure ou égale à 24, utilisez REAL4. Si la précision est supérieure à 24, utilisez REAL8.
NUMBER (P,S)	Lorsque l'échelle est supérieure à 0, utilisez NUMERIC. Lorsque l'échelle est 0 : • Lorsque la précision est inférieure ou égale à 2, utilisez INT1. • Et que la précision est supérieure à 2 et inférieure ou égale à 4, utilisez INT2.

Type de données Oracle	AWS DMS type de données
	• Et que la précision est supérieure à 4 et inférieure ou égale à 9, utilisez INT4. • Et que la précision est supérieure à 9, utilisez NUMERIC. • Et que la précision est supérieure ou égale à l'échelle, utilisez NUMERIC. Lorsque l'échelle est inférieure à 0, utilisez REAL8.
DATE	DATETIME
INTERVAL_YEAR TO MONTH	STRING (avec indication d'intervalle year_to_month)
INTERVAL_DAY TO SECOND	STRING (avec indication d'intervalle day_to_second)
TIMESTAMP	DATETIME
TIMESTAMP WITH TIME ZONE	STRING (avec indication timestamp_with_timezone)
TIMESTAMP WITH LOCAL TIME ZONE	STRING (avec indication timestamp_with_local_timezone)
CHAR	CHAÎNE
VARCHAR2	CHAÎNE
NCHAR	WSTRING
NVARCHAR2	WSTRING
RAW	BYTES
REAL	REAL8

Type de données Oracle	AWS DMS type de données
BLOB	BLOB Pour utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation des types de données BLOB pour une tâche spécifique. AWS DMS prend en charge les types de données BLOB uniquement dans les tables qui incluent une clé primaire.
CLOB	CLOB Pour utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation des types de données CLOB pour une tâche spécifique. Pendant le CDC, AWS DMS prend en charge les types de données CLOB uniquement dans les tables qui incluent une clé primaire.
NCLOB	NCLOB Pour utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation des types de données NCLOB pour une tâche spécifique. Pendant le CDC, AWS DMS prend en charge les types de données NCLOB uniquement dans les tables qui incluent une clé primaire.

Type de données Oracle	AWS DMS type de données
LONG	CLOB Le type de données LONG n'est pas pris en charge en mode d'application optimisé par lots (mode TurboStream CDC). Pour utiliser ce type de données avec AWS DMS, activez l'utilisation de LOB pour une tâche spécifique. Pendant le CDC ou le chargement complet, AWS DMS prend en charge les types de données LOB uniquement dans les tables dotées d'une clé primaire. De plus, AWS DMS ne prend pas en charge le mode LOB complet pour le chargement de colonnes LONGUES. Vous pouvez plutôt utiliser le mode LOB limité pour migrer les colonnes LONG vers une cible Oracle. En mode LOB limité, AWS DMS tronque à 64 Ko toutes les données que vous définissez comme des colonnes LONGUES de plus de 64 Ko. Pour plus d'informations sur le support LOB dans AWS DMS, voir Configuration du support LOB pour les bases de données sources dans une tâche AWS DMS

Type de données Oracle	AWS DMS type de données
LONG RAW	BLOB Le type de données LONG RAW n'est pas pris en charge en mode d'application optimisé par lots (mode TurboStream CDC). Pour utiliser ce type de données avec AWS DMS, activez l'utilisa tion de LOB pour une tâche spécifique. Pendant le CDC ou le chargement complet, AWS DMS prend en charge les types de données LOB uniquement dans les tables dotées d'une clé primaire. De plus, AWS DMS ne prend pas en charge le mode LOB complet pour le chargement de colonnes LONG RAW. Vous pouvez plutôt utiliser le mode LOB limité pour migrer les colonnes LONG RAW vers une cible Oracle. En mode LOB limité, AWS DMS tronque à 64 Ko toutes les données que vous définissez comme des colonnes LONG RAW de plus de 64 Ko. Pour plus d'informations sur le support LOB dans AWS DMS, voir Configuration du support LOB pour les bases de données sources dans une tâche AWS DMS
XMLTYPE	CLOB
SDO_GEOMETRY	BLOB (pour une migration Oracle vers Oracle) CLOB (pour une migration Oracle vers PostgreSQL)

Les tables Oracle utilisées comme source avec des colonnes des types de données suivants ne sont pas prises en charge et ne peuvent pas être répliquées. La réplication des colonnes avec ces types de données a pour conséquence une colonne nulle.

- BFILE
- ROWID
- REF
- UROWID
- Types de données définis par l'utilisateur

- ANYDATA
- VARRAY

> **Note**
>
> Les colonnes virtuelles ne sont pas prises en charge.

Migration de types de données spatiales Oracle

Les données spatiales identifient les informations de géométrie d'un objet ou d'un emplacement dans l'espace. Dans une base de données Oracle, la description géométrique d'un objet spatial est stockée dans un objet de type SDO_GEOMETRY. Dans cet objet, la description géométrique est stockée dans une seule ligne dans une seule colonne d'une table définie par l'utilisateur.

AWS DMS prend en charge la migration du type Oracle SDO_GEOMETRY d'une source Oracle vers une cible Oracle ou PostgreSQL.

Lorsque vous migrez des types de données spatiales Oracle à l'aide de AWS DMS, tenez compte des points suivants :

- Lors de la migration vers une cible Oracle, assurez-vous de transférer manuellement les entrées USER_SDO_GEOM_METADATA qui incluent des informations de type.
- Lors de la migration d'un point de terminaison source Oracle vers un point de terminaison AWS DMS cible PostgreSQL, crée des colonnes cibles. Ces colonnes contiennent des informations sur la géométrie et le type de géographie par défaut avec une dimension 2D et un identificateur de référence spatiale (SRID) égal à zéro (0). Par exemple : `GEOMETRY, 2, 0`.
- Pour les sources Oracle version 12.1 ou antérieures qui migrent vers des cibles PostgreSQL, convertissez les objets `SDO_GEOMETRY` au format `GEOJSON` à l'aide de la fonction `SDO2GEOJSON` ou de l'attribut de connexion `spatialSdo2GeoJsonFunctionName` supplémentaire. Pour plus d'informations, consultez Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS.
- AWS DMS prend en charge les migrations de colonnes spatiales Oracle pour le mode LOB complet uniquement. AWS DMS ne prend pas en charge les modes LOB limité ou LOB en ligne. Pour plus d'informations sur le mode LOB, consultez Configuration du support LOB pour les bases de données sources dans une tâche AWS DMS.

- Comme le mode Full LOB est AWS DMS uniquement compatible avec la migration des colonnes spatiales Oracle, la table des colonnes a besoin d'une clé primaire et d'une clé unique. Si la table n'a pas de clé primaire et de clé unique, elle est ignorée de la migration.

Utilisation d'une base de données Microsoft SQL Server comme source pour AWS DMS

Migrez les données d'une ou de plusieurs bases de données Microsoft SQL Server à l'aide de AWS DMS. Avec une base de données SQL Server comme source, vous pouvez migrer des données vers une autre base de données SQL Server ou vers l'une des autres bases de données AWS DMS prises en charge.

Pour plus d'informations sur les versions de SQL Server prises AWS DMS en charge en tant que source, consultezSources pour AWS DMS.

La base de données source SQL Server peut être installée sur n'importe quel ordinateur dans votre réseau. Un compte SQL Server avec les privilèges d'accès appropriés à la base de données source pour le type de tâche que vous avez choisi est également nécessaire pour une utilisation avec AWS DMS. Ce compte doit disposer des autorisations `view definition` et `view server state`. Vous ajoutez cette autorisation à l'aide de la commande suivante :

```
grant view definition to [user]
grant view server state to [user]
```

AWS DMS prend en charge la migration de données à partir d'instances nommées de SQL Server. Vous pouvez utiliser la notation suivante dans le nom du serveur lorsque vous créez le point de terminaison source.

```
IPAddress\InstanceName
```

Par exemple, voici un nom de serveur correct de point de terminaison source. Ici, la première partie du nom est l'adresse IP du serveur et la seconde partie est le nom de l'instance SQL Server (dans cet exemple, SQLTest).

```
10.0.0.25\SQLTest
```

Obtenez également le numéro de port sur lequel votre instance nommée de SQL Server écoute et utilisez-le pour configurer votre point de terminaison AWS DMS source.

> **Note**
>
> Le port 1433 est le port par défaut pour Microsoft SQL Server. Mais les ports dynamiques qui changent chaque fois que SQL Server est démarré, et les numéros de port statiques spécifiques utilisés pour se connecter à SQL Server via un pare-feu sont également souvent utilisés. Vous souhaitez donc connaître le numéro de port réel de votre instance nommée de SQL Server lorsque vous créez le point de terminaison AWS DMS source.

Vous pouvez utiliser SSL pour chiffrer les connexions entre votre point de terminaison SQL Server et l'instance de réplication. Pour plus d'informations sur l'utilisation de SSL avec un point de terminaison SQL Server, consultez Utilisation du protocole SSL avec AWS Database Migration Service.

Pour plus de détails sur l'utilisation des bases de données source SQL Server AWS DMS, consultez les sections suivantes.

Rubriques

- Limitations relatives à l'utilisation de SQL Server comme source pour AWS DMS
- Autorisations pour les tâches de chargement complet uniquement
- Conditions préalables pour l'utilisation de la réplication continue (CDC) à partir d'une source SQL Server
- Capture des modifications de données pour SQL Server autogéré sur site ou sur Amazon EC2
- Configuration de la réplication continue sur une instance de base de données SQL Server cloud
- Paramètres recommandés lors de l'utilisation d'Amazon RDS for SQL Server comme source pour AWS DMS
- Méthodes de compression prises en charge pour SQL Server
- Utilisation de groupes de AlwaysOn disponibilité SQL Server autogérés
- Exigences de sécurité lors de l'utilisation de SQL Server comme source pour AWS Database Migration Service
- Paramètres du point de terminaison lors de l'utilisation de SQL Server comme source pour AWS DMS
- Types de données sources pour SQL Server

Limitations relatives à l'utilisation de SQL Server comme source pour AWS DMS

Les limites suivantes s'appliquent lors de l'utilisation d'une base de données SQL Server comme source pour AWS DMS :

- La propriété d'identité d'une colonne n'est pas migrée vers une colonne de base de données cible.
- Le point de terminaison SQL Server ne prend pas en charge l'utilisation de tables contenant des colonnes éparses.
- L'authentification Windows n'est pas prise en charge.
- Les modifications apportées aux champs calculés dans un serveur SQL Server ne sont pas répliquées.
- Les tables temporelles ne sont pas prises en charge.
- L'échange de partition SQL Server n'est pas pris en charge.
- Lorsque vous utilisez les utilitaires WRITETEXT et UPDATETEXT, AWS DMS ne capture pas les événements appliqués à la base de données source.
- Le modèle de langage de manipulation de données (DML) suivant n'est pas pris en charge.

```
SELECT * INTO new_table FROM existing_table
```

- Lorsque vous utilisez SQL Server comme source, le chiffrement au niveau des colonnes n'est pas pris en charge.
- AWS DMS ne prend pas en charge les audits au niveau du serveur sur SQL Server 2008 ou SQL Server 2008 R2 en tant que sources. Cela est dû à un problème connu avec SQL Server 2008 et 2008 R2. Par exemple, l'exécution de la commande suivante AWS DMS entraîne un échec.

```
USE [master]
GO
ALTER SERVER AUDIT [my_audit_test-20140710] WITH (STATE=on)
GO
```

- Les colonnes de géométrie ne sont pas prises en charge en mode LOB complet si SQL Server est utilisé en tant que source. Utilisez plutôt le mode LOB limité ou définissez le paramètre de tâche `InlineLobMaxSize` pour utiliser le mode LOB en ligne.
- Lorsque vous utilisez une base de données source Microsoft SQL Server dans une tâche de réplication, les définitions du diffuseur de publication de réplication SQL Server ne sont pas supprimées si vous supprimez la tâche. Un administrateur système Microsoft SQL Server doit supprimer ces définitions de Microsoft SQL Server.

- La migration des données depuis les non-schema-bound vues et les données liées au schéma est prise en charge uniquement pour les tâches à chargement complet.
- La modification de noms de tables à l'aide de sp_rename n'est pas prise en charge (par exemple, `sp_rename 'Sales.SalesRegion', 'SalesReg;)`
- La modification de noms de colonnes à l'aide de sp_rename n'est pas prise en charge (par exemple, `sp_rename 'Sales.Sales.Region', 'RegID', 'COLUMN';)`
- AWS DMS ne prend pas en charge le traitement des modifications pour définir ou annuler les valeurs par défaut des colonnes (en utilisant la `ALTER COLUMN SET DEFAULT` clause avec `ALTER TABLE` instructions).
- AWS DMS ne prend pas en charge le traitement des modifications pour définir la nullité des colonnes (en utilisant la `ALTER COLUMN [SET|DROP] NOT NULL` clause avec des `ALTER TABLE` instructions).
- Avec SQL Server 2012 et SQL Server 2014, lors de l'utilisation de la réplication DMS avec des groupes de disponibilité, la base de données de distribution ne peut pas être placée dans un groupe de disponibilité. SQL 2016 prend en charge le placement de la base de données de distribution dans un groupe de disponibilité, à l'exception des bases de données de distribution utilisées dans les topologies de fusion, bidirectionnelles ou de peer-to-peer réplication.
- Pour les tables partitionnées, AWS DMS ne prend pas en charge les différents paramètres de compression des données pour chaque partition.
- Lorsque vous insérez une valeur dans les types de données spatiales SQL Server (GEOGRAPHY et GEOMETRY), vous pouvez ignorer la propriété SRID (Spatial Reference System Identifier) ou spécifier un nombre différent. Lors de la réplication de tables contenant des types de données spatiales, AWS DMS remplace le SRID par le SRID par défaut (0 pour GEOMETRY et 4326 pour GEOGRAPHY).
- Si votre base de données n'est pas configurée pour MS-REPLICATION ou MS-CDC, vous pouvez toujours capturer des tables qui n'ont pas de clé primaire, mais seuls les événements DML INSERT/DELETE sont capturés. Les événements UPDATE et TRUNCATE TABLE sont ignorés.
- Les index Columnstore ne sont pas pris en charge.
- Les tables optimisées pour la mémoire (utilisant OLTP en mémoire) ne sont pas prises en charge.
- Lors de la réplication d'une table avec une clé primaire composée de plusieurs colonnes, la mise à jour des colonnes de clé primaire pendant le chargement complet n'est pas prise en charge.
- La durabilité retardée n'est pas prise en charge.

- Le paramètre de point de terminaison `readBackup0nly=Y` (attribut de connexion supplémentaire) ne fonctionne pas sur les instances source RDS for SQL Server en raison de la manière dont RDS effectue les sauvegardes.
- `EXCLUSIVE_AUTOMATIC_TRUNCATION` ne fonctionne pas sur les instances source Amazon RDS SQL Server, car les utilisateurs de RDS ne sont pas autorisés à exécuter la procédure stockée SQL Server `sp_repldone`.
- AWS DMS ne capture pas les commandes de troncation.
- AWS DMS ne prend pas en charge la réplication à partir de bases de données lorsque la restauration accélérée des bases de données (ADR) est activée.
- AWS DMS ne prend pas en charge la capture d'instructions en langage de définition de données (DDL) et en langage de manipulation de données (DML) au cours d'une seule transaction.
- AWS DMS ne prend pas en charge la réplication des packages d'applications au niveau des données (DACPAC).
- Les instructions UPDATE qui impliquent des clés primaires ou des index uniques et qui mettent à jour plusieurs lignes de données peuvent provoquer des conflits lorsque vous appliquez les modifications à la base de données cible. Cela peut se produire, par exemple, lorsque la base de données cible applique des mises à jour sous forme d'instructions INSERT et DELETE au lieu d'une seule instruction UPDATE. Avec le mode d'application optimisé par lots, la table peut être ignorée. Avec le mode d'application transactionnel, l'opération UPDATE peut entraîner des violations de contrainte. Pour éviter ce problème, rechargez la table correspondante. Vous pouvez également rechercher les enregistrements problématiques dans la table de contrôle Application des exceptions (`dmslogs.awsdms_apply_exceptions`) et les modifier manuellement dans la base de données cible. Pour de plus amples informations, veuillez consulter Paramètres de réglage du traitement des modifications.
- AWS DMS ne prend pas en charge la réplication de tables et de schémas dont le nom inclut un caractère spécial de l'ensemble suivant.

  ```
  \\ -- \n \" \b \r ' \t ;
  ```

- Le masquage des données n'est pas pris en charge. AWS DMS migre les données masquées sans les masquer.
- AWS DMS réplique jusqu'à 32 767 tables avec des clés primaires et jusqu'à 1 000 colonnes pour chaque table. Cela est dû au fait que AWS DMS crée un article de réplication SQL Server pour chaque table répliquée, et les articles de réplication de SQL Server présentent ces limites.

- Lorsque vous utilisez la capture des données de modification (CDC), vous devez définir toutes les colonnes qui constituent un index unique en tant que `NOT NULL`. Si cette exigence n'est pas remplie, l'erreur système SQL Server 22838 se produit.

Les limitations suivantes s'appliquent lors de l'accès aux journaux de transactions de sauvegarde :

- Les sauvegardes chiffrées ne sont pas prises en charge.
- Les sauvegardes stockées sur une URL ou sur Windows Azure ne sont pas prises en charge.
- AWS DMS ne prend pas en charge le traitement direct des sauvegardes du journal des transactions au niveau des fichiers à partir de dossiers partagés alternatifs.

Autorisations pour les tâches de chargement complet uniquement

Les autorisations suivantes sont requises pour effectuer des tâches de chargement complet uniquement. Notez que AWS DMS cela ne crée pas le `dms_user` login. Pour en savoir plus sur la création d'un identifiant de connexion pour SQL Server, consultez Création d'un utilisateur de base de données avec Microsoft SQL Server.

```
USE db_name;

                CREATE USER dms_user FOR LOGIN dms_user;
                ALTER ROLE [db_datareader] ADD MEMBER dms_user;
                GRANT VIEW DATABASE STATE to dms_user ;

                USE master;

                GRANT VIEW SERVER STATE TO dms_user;
```

Conditions préalables pour l'utilisation de la réplication continue (CDC) à partir d'une source SQL Server

Vous pouvez utiliser la réplication continue (capture des données de modification ou CDC) pour une base de données SQL Server autogérée sur site ou sur Amazon EC2, ou une base de données de cloud telle que Amazon RDS ou une instance gérée par Microsoft Azure SQL.

Les conditions suivantes s'appliquent précisément lorsque la réplication continue est utilisée avec une base de données SQL Server comme source pour AWS DMS :

- SQL Server doit être configuré pour des sauvegardes complètes et vous devez effectuer une sauvegarde avant de commencer la réplication des données.
- Le modèle de récupération doit être défini sur Bulk logged ou Full.
- La sauvegarde SQL Server sur plusieurs disques n'est pas prise en charge. Si la sauvegarde est définie pour écrire la sauvegarde de la base de données dans plusieurs fichiers sur différents disques, AWS DMS vous ne pouvez pas lire les données et la AWS DMS tâche échoue.
- Pour les sources SQL Server autogérées, les définitions du diffuseur de publication de réplication SQL Server pour la source utilisée dans une tâche de CDC DMS ne sont pas supprimées lorsque vous supprimez la tâche. Un administrateur système SQL Server doit supprimer ces définitions de SQL Server pour les sources autogérées.
- Pendant le CDC, il AWS DMS doit consulter les sauvegardes du journal des transactions de SQL Server pour lire les modifications. AWS DMS ne prend pas en charge les sauvegardes du journal des transactions SQL Server créées à l'aide de logiciels de sauvegarde tiers qui ne sont pas au format natif. Pour prendre en charge les sauvegardes du journal des transactions qui sont au format natif et qui ont été créées à l'aide d'un logiciel de sauvegarde tiers, ajoutez l'attribut de connexion `use3rdPartyBackupDevice=Y` au point de terminaison source.
- Pour les sources SQL Server autogérées, sachez que SQL Server ne capture pas les modifications sur les tables nouvellement créées tant qu'elles n'ont pas été publiées. Lorsque des tables sont ajoutées à une source SQL Server, AWS DMS gère la création de la publication. Toutefois, ce processus peut prendre plusieurs minutes. Les opérations réalisées sur les tables nouvellement créées pendant ce délai ne sont pas capturées ni répliquées sur la cible.
- AWS DMS la capture des données de modification nécessite l'activation de la journalisation complète des transactions dans SQL Server. Pour activer la journalisation complète des transactions dans SQL Server, activez MS-REPLICATION ou CHANGE DATA CAPTURE (CDC).
- Les entrées tlog SQL Server ne seront pas marquées pour être réutilisées tant que la tâche de capture MS CDC n'aura pas traité ces modifications.
- Les opérations CDC ne sont pas prises en charge sur les tables de mémoire optimisée. Cette limitation s'applique à SQL Server 2014 (lorsque la fonctionnalité a été introduite pour la première fois) et aux versions ultérieures.
- AWS DMS la capture des données de modification nécessite une base de données de distribution par défaut sur Amazon EC2 ou On-Prem SQL Server comme source. Vous devez donc vous assurer d'avoir activé le distributeur lors de la configuration de la réplication MS pour les tables comportant des clés primaires.

Capture des modifications de données pour SQL Server autogéré sur site ou sur Amazon EC2

Pour capturer les modifications à partir d'une base de données Microsoft SQL Server source, assurez-vous que la base de données est configurée pour des sauvegardes complètes. Configurez la base de données en mode récupération complète ou en mode journalisation en bloc.

Pour une source SQL Server autogérée, AWS DMS utilise ce qui suit :

Réplication Microsoft

Pour capturer les modifications pour des tables comportant des clés primaires. Vous pouvez le configurer automatiquement en octroyant des privilèges d'administrateur système à l'utilisateur du AWS DMS point de terminaison sur l'instance SQL Server source. Vous pouvez également suivre les étapes décrites dans cette section pour préparer la source et utiliser un utilisateur ne disposant pas des privilèges d'administrateur système pour le AWS DMS point de terminaison.

MS-CDC

Pour capturer les modifications pour des tables sans clés primaires. Activez MS-CDC au niveau de la base de données et individuellement pour toutes les tables.

Lorsque vous configurez une base de données SQL Server pour la réplication continue (CDC), vous pouvez procéder de différentes manières :

- Configurez la réplication continue en utilisant le rôle sysadmin.
- Configurez la réplication continue pour qu'elle n'utilise pas le rôle sysadmin.

Configuration de la réplication continue sur une instance SQL Server autogérée

Cette section contient des informations sur la configuration de la réplication continue sur une instance SQL Server autogérée en utilisant ou non le rôle sysadmin.

Rubriques

- Configuration de la réplication continue sur une instance SQL Server autogérée : Utilisation du rôle sysadmin
- Configuration de la réplication continue sur une instance SQL Server autonome : sans le rôle sysadmin

Configuration de la réplication continue sur une instance SQL Server autogérée : Utilisation du rôle sysadmin

AWS DMS la réplication continue pour SQL Server utilise la réplication native de SQL Server pour les tables dotées de clés primaires et la capture des données de modification (CDC) pour les tables sans clés primaires.

Avant de configurer la réplication continue, consultez Conditions préalables pour l'utilisation de la réplication continue (CDC) à partir d'une source SQL Server.

Pour les tables avec des clés primaires, AWS DMS vous pouvez généralement configurer les artefacts requis sur la source. En revanche, pour les instances SQL Server source autogérées, assurez-vous de configurer manuellement la distribution SQL Server en premier. Ensuite, les utilisateurs de la AWS DMS source disposant d'une autorisation d'administrateur système peuvent créer automatiquement la publication pour les tables avec des clés primaires.

Pour vérifier si la distribution a déjà été configurée, exécutez la commande suivante.

```
sp_get_distributor
```

Si le résultat est `NULL` pour la distribution de colonnes, cela signifie que la distribution n'est pas configurée. Vous pouvez procéder comme suit pour configurer la distribution.

Pour configurer la distribution

1. Connectez-vous à la base de données source SQL Server à l'aide de l'outil SQL Server Management Studio (SSMS).
2. Ouvrez le menu contextuel (clic droit) pour le dossier Replication, puis choisissez Configurer la distribution. L'assistant Configurer la distribution s'affiche.
3. Suivez les instructions de l'assistant pour entrer les valeurs par défaut et créer la distribution.

Pour configurer la CDC

AWS DMS la version 3.4.7 et les versions ultérieures peuvent configurer MS CDC pour votre base de données et toutes vos tables automatiquement si vous n'utilisez pas de réplique en lecture seule. Pour utiliser cette fonctionnalité, définissez l'ECA `SetUpMsCdcForTables` sur true. Pour en savoir plus sur les ECA, consultez Paramètres de point de terminaison.

Pour les versions AWS DMS antérieures à 3.4.7, ou pour une réplique en lecture seule en tant que source, effectuez les opérations suivantes :

1. Pour les tables sans clés primaires, configurez MS-CDC pour la base de données. Pour ce faire, utilisez un compte auquel le rôle sysadmin est affecté et exécutez la commande suivante.

   ```
   use [DBname]
   EXEC sys.sp_cdc_enable_db
   ```

2. Configurez ensuite MS-CDC pour chacune des tables sources. Pour chaque table disposant de clés uniques mais ne disposant pas de clé primaire, exécutez la requête suivante afin de configurer MS-CDC.

   ```
   exec sys.sp_cdc_enable_table
   @source_schema = N'schema_name',
   @source_name = N'table_name',
   @index_name = N'unique_index_name',
   @role_name = NULL,
   @supports_net_changes = 1
   GO
   ```

3. Pour chaque table ne disposant pas de clé primaire ou de clé unique, exécutez la requête suivante afin de configurer MS-CDC.

   ```
   exec sys.sp_cdc_enable_table
   @source_schema = N'schema_name',
   @source_name = N'table_name',
   @role_name = NULL
   GO
   ```

Pour plus d'informations sur la configuration de MS-CDC pour des tables spécifiques, consultez la documentation SQL Server.

Configuration de la réplication continue sur une instance SQL Server autonome : sans le rôle sysadmin

Pour en savoir plus sur la configuration de la réplication continue sur une instance SQL Server autonome sans le rôle sysadmin, consultez Configuration de la réplication continue sur une instance SQL Server autonome : sans le rôle sysadmin.

Configuration de la réplication continue sur une instance de base de données SQL Server cloud

Cette section explique comment configurer la CDC sur une instance de base de données SQL Server hébergée sur le cloud. Une instance SQL Server hébergée sur le cloud est une instance exécutée sur Amazon RDS for SQL Server, une instance gérée par Azure SQL ou toute autre instance SQL Server gérée dans le cloud. Pour en savoir plus sur les limitations relatives à la réplication continue pour chaque type de base de données, consultez Limitations relatives à l'utilisation de SQL Server comme source pour AWS DMS.

Avant de configurer la réplication continue, consultez Conditions préalables pour l'utilisation de la réplication continue (CDC) à partir d'une source SQL Server.

À la différence des sources Microsoft SQL Server autogérées, Amazon RDS for SQL Server ne prend pas en charge la réplication Microsoft. Par conséquent, AWS DMS vous devez utiliser MS-CDC pour les tables avec ou sans clés primaires.

Amazon RDS n'accorde pas de privilèges d'administrateur système pour configurer les artefacts de réplication AWS DMS utilisés pour les modifications continues dans une instance SQL Server source. Veillez à activer MS-CDC pour l'instance Amazon RDS (en utilisant des privilèges d'utilisateur principal) comme décrit dans la procédure suivante.

Pour activer MS-CDC pour une instance de base de données SQL Server cloud

1. Exécutez l'une des requêtes suivantes au niveau de la base de données.

   Pour une instance de base de données RDS for SQL Server, utilisez cette requête.

   ```
   exec msdb.dbo.rds_cdc_enable_db 'DB_name'
   ```

   Pour une instance de base de données gérée par Azure SQL, utilisez cette requête.

   ```
   USE DB_name
   GO
   EXEC sys.sp_cdc_enable_db
   GO
   ```

2. Pour chaque table disposant d'une clé primaire, exécutez la requête suivante afin d'activer MS-CDC.

```
exec sys.sp_cdc_enable_table
@source_schema = N'schema_name',
@source_name = N'table_name',
@role_name = NULL,
@supports_net_changes = 1
GO
```

Pour chaque table disposant de clés uniques mais ne disposant pas de clé primaire, exécutez la requête suivante afin d'activer MS-CDC.

```
exec sys.sp_cdc_enable_table
@source_schema = N'schema_name',
@source_name = N'table_name',
@index_name = N'unique_index_name',
@role_name = NULL,
@supports_net_changes = 1
GO
```

Pour chaque table ne disposant pas de clé unique ni de clé primaire, exécutez la requête suivante afin d'activer MS-CDC.

```
exec sys.sp_cdc_enable_table
@source_schema = N'schema_name',
@source_name = N'table_name',
@role_name = NULL
GO
```

3. Définissez la période de rétention pour rendre les modifications disponibles au niveau de la source en utilisant les commandes suivantes.

```
use dbname
EXEC sys.sp_cdc_change_job @job_type = 'capture' ,@pollinginterval = 86399
exec sp_cdc_stop_job 'capture'
exec sp_cdc_start_job 'capture'
```

Le paramètre `@pollinginterval` est mesuré en secondes avec une valeur recommandée définie sur 86399. Cela signifie que le journal des transactions conserve les modifications pendant 86 399 secondes (un jour) lorsque `@pollinginterval = 86399`. La procédure `exec sp_cdc_start_job 'capture'` lance les paramètres.

> **Note**
>
> Dans certaines versions de SQL Server, si la valeur de `pollinginterval` est définie sur plus de 3 599 secondes, elle est réinitialisée à la valeur par défaut de 5 secondes. Dans ce cas, les entrées du T-Log sont purgées avant de AWS DMS pouvoir les lire. Pour déterminer quelles versions de SQL Server sont concernées par ce problème connu, consultez cet article de la base de connaissances Microsoft.

Si vous utilisez Amazon RDS avec le mode multi-AZ, veillez à configurer également votre instance secondaire de sorte qu'elle ait les bonnes valeurs en cas de basculement.

```
exec rdsadmin..rds_set_configuration 'cdc_capture_pollinginterval' , 86399
```

Si une tâche de AWS DMS réplication qui capture les modifications continues apportées à votre source SQL Server s'arrête pendant plus d'une heure, suivez la procédure ci-dessous.

Pour maintenir la période de rétention pendant une tâche AWS DMS de réplication

1. Arrêtez la tâche qui tronque les journaux de transactions à l'aide de la commande suivante.

   ```
   exec sp_cdc_stop_job 'capture'
   ```

2. Trouvez votre tâche sur la AWS DMS console et reprenez-la.
3. Choisissez l'onglet Surveillance, puis vérifiez la métrique `CDCLatencySource`.
4. Une fois que la métrique `CDCLatencySource` est égale à 0 (zéro) et ne change plus, redémarrez la tâche qui tronque les journaux de transactions à l'aide de la commande suivante.

   ```
   exec sp_cdc_start_job 'capture'
   ```

N'oubliez pas de démarrer la tâche qui tronque les journaux de transactions SQL Server. Dans le cas contraire, l'espace de stockage sur votre instance SQL Server risque de se remplir.

Limitations de la réplication continue sur une instance de base de données SQL Server cloud

- AWS DMS prend en charge la réplication continue (CDC) uniquement avec le journal des transactions actif. Vous ne pouvez pas utiliser le journal de sauvegarde avec la CDC.
- Vous risquez de perdre des événements si vous les déplacez du journal des transactions actif vers le journal de sauvegarde ou si vous les tronquez du journal des transactions actif.

Paramètres recommandés lors de l'utilisation d'Amazon RDS for SQL Server comme source pour AWS DMS

Lorsque vous utilisez Amazon RDS for SQL Server en tant que source, la tâche de capture repose sur les paramètres `maxscans` et `maxtrans`. Ces paramètres régissent le nombre maximal d'analyses effectuées par la capture dans le journal des transactions et le nombre de transactions traitées pour chaque analyse.

Pour les bases de données où le nombre de transactions est supérieur à `maxtrans*maxscans`, l'augmentation de la valeur `polling_interval` peut entraîner une accumulation d'enregistrements dans le journal des transactions actif. À son tour, cette accumulation peut entraîner une augmentation de la taille du journal des transactions.

Notez que cela AWS DMS ne repose pas sur la tâche de capture MS-CDC. La tâche de capture MS-CDC marque les entrées du journal des transactions comme ayant été traitées. Cela permet à la tâche de sauvegarde du journal des transactions de supprimer les entrées du journal des transactions.

Nous vous recommandons de surveiller la taille du journal des transactions et la réussite des tâches MS-CDC. Si les tâches MS-CDC échouent, le journal des transactions peut augmenter de manière excessive et provoquer des échecs de AWS DMS réplication. Vous pouvez surveiller les erreurs des tâches de capture MS-CDC à l'aide de la vue de gestion dynamique `sys.dm_cdc_errors` de la base de données source. Vous pouvez surveiller la taille du journal des transactions à l'aide de la commande de gestion `DBCC SQLPERF(LOGSPACE)`.

Pour remédier à l'augmentation du journal des transactions provoquée par MS-CDC

1. Vérifiez le `Log Space Used %` pour lequel la base de données AWS DMS est en train de se répliquer et vérifiez qu'il augmente continuellement.

   ```
   DBCC SQLPERF(LOGSPACE)
   ```

2. Identifiez ce qui bloque le processus de sauvegarde du journal des transactions.

```
Select log_reuse_wait, log_reuse_wait_desc, name from sys.databases where name =
 db_name();
```

   Si la valeur `log_reuse_wait_desc` est égale à `REPLICATION`, la conservation de la sauvegarde du journal est due à la latence dans MS-CDC.

3. Augmentez le nombre d'événements traités par la tâche de capture en augmentant les valeurs des paramètres `maxtrans` et `maxscans`.

```
EXEC sys.sp_cdc_change_job @job_type = 'capture' ,@maxtrans = 5000, @maxscans = 20
exec sp_cdc_stop_job 'capture'
exec sp_cdc_start_job 'capture'
```

Pour résoudre ce problème, définissez les valeurs de `maxscans` et de `maxtrans` manière à ce `maxtrans*maxscans` qu'elles soient égales au nombre moyen d'événements générés pour les tables AWS DMS répliquées à partir de la base de données source chaque jour.

Si vous définissez ces paramètres sur une valeur supérieure à la valeur recommandée, les tâches de capture traitent tous les événements des journaux de transactions. Si vous définissez ces paramètres sur une valeur inférieure à la valeur recommandée, la latence MS-CDC augmente ainsi que la taille de votre journal des transactions.

L'identification des valeurs appropriées pour `maxscans` et `maxtrans` peut s'avérer difficile, car les modifications de la charge de travail produisent un nombre variable d'événements. Dans ce cas, nous vous recommandons de configurer la surveillance sur la latence MS-CDC. Pour plus d'informations, consultez Monitor the process dans la documentation de SQL Server. Configurez ensuite `maxtrans` et `maxscans` de manière dynamique en fonction des résultats de la surveillance.

Si la AWS DMS tâche ne parvient pas à trouver les numéros de séquence de journal (LSN) nécessaires pour reprendre ou poursuivre la tâche, celle-ci peut échouer et nécessiter un rechargement complet.

> **Note**
>
> Lors de l'utilisation AWS DMS pour répliquer des données à partir d'une source RDS pour SQL Server, vous pouvez rencontrer des erreurs lorsque vous tentez de reprendre la réplication après un événement stop-start de l'instance Amazon RDS. Cela est dû au fait que

le processus SQL Server Agent redémarre le processus de capture lorsqu'il redémarre après l'événement d'arrêt/démarrage. Cela permet de contourner l'intervalle d'interrogation MS-CDC.
De ce fait, sur les bases de données dont le volume de transactions est inférieur au traitement de la tâche de capture MS-CDC, les données peuvent être traitées ou marquées comme répliquées et sauvegardées avant de AWS DMS pouvoir reprendre là où elles s'étaient arrêtées, ce qui entraîne l'erreur suivante :

```
[SOURCE_CAPTURE ]E: Failed to access LSN '0000dbd9:0006f9ad:0003' in
 the backup log sets since BACKUP/LOG-s are not available. [1020465]
 (sqlserver_endpoint_capture.c:764)
```

Pour atténuer ce problème, définissez les valeurs `maxtrans` et `maxscans` comme recommandé précédemment.

Méthodes de compression prises en charge pour SQL Server

Notez ce qui suit concernant la prise en charge des méthodes de compression SQL Server dans AWS DMS :

- AWS DMS prend en charge la compression ligne/page dans SQL Server version 2008 et versions ultérieures.
- AWS DMS ne prend pas en charge le format de stockage Vardecimal.
- AWS DMS ne prend pas en charge les colonnes clairsemées et la compression de structure en colonnes.

Utilisation de groupes de AlwaysOn disponibilité SQL Server autogérés

Les groupes de disponibilité SQL Server Always On assurent la haute disponibilité et la reprise après sinistre comme alternative au niveau de l'entreprise à la mise en miroir de base de données.

Dans AWS DMS, vous pouvez migrer les modifications à partir d'une seule réplique de groupe de disponibilité principal ou secondaire.

Utilisation du réplica de groupe de disponibilité principal

Pour utiliser le groupe de disponibilité principal comme source dans AWS DMS, procédez comme suit :

1. Activez l'option de distribution pour toutes les instances de SQL Server dans vos réplicas de disponibilité. Pour de plus amples informations, veuillez consulter Configuration de la réplication continue sur une instance SQL Server autogérée.
2. Dans la AWS DMS console, ouvrez les paramètres de la base de données source SQL Server. Pour Nom du serveur, spécifiez le nom DNS (Domain Name Service) ou l'adresse IP qui a été configuré(e) pour l'écouteur du groupe de disponibilité.

Lorsque vous démarrez une AWS DMS tâche pour la première fois, le démarrage peut prendre plus de temps que d'habitude. Cette lenteur est due au fait que la création des articles de la table est dupliquée par le serveur du groupe de disponibilité.

Utilisation d'un réplica de groupe de disponibilité secondaire

Pour utiliser un groupe de disponibilité secondaire comme source dans AWS DMS, procédez comme suit :

1. Utilisez les mêmes informations d'identification pour vous connecter à des répliques individuelles que celles utilisées par l'utilisateur du point de terminaison AWS DMS source.
2. Assurez-vous que votre instance de AWS DMS réplication peut résoudre les noms DNS de toutes les répliques existantes et s'y connecter. Vous pouvez utiliser la requête SQL suivante pour obtenir les noms DNS de tous vos réplicas.

   ```
   select ar.replica_server_name, ar.endpoint_url from sys.availability_replicas ar
   JOIN sys.availability_databases_cluster adc
   ON adc.group_id = ar.group_id AND adc.database_name = '<source_database_name>';
   ```

3. Lorsque vous créez le point de terminaison source, spécifiez le nom DNS de l'écouteur du groupe de disponibilité pour le nom du serveur du point de terminaison ou pour l'adresse du serveur du secret de point de terminaison. Pour plus d'informations sur les écouteurs de groupe de disponibilité, consultez Qu'est-ce qu'un écouteur de groupe de disponibilité ? dans la documentation de SQL Server.

   Vous pouvez utiliser un serveur DNS public ou un serveur DNS sur site pour résoudre l'écouteur du groupe de disponibilité, le réplica principal et les réplicas secondaires. Pour utiliser un serveur

DNS sur site, configurez Amazon Route 53 Resolver. Pour de plus amples informations, veuillez consulter Utilisation de votre propre serveur de noms sur site.

4. Ajoutez les attributs de connexion supplémentaires suivants au point de terminaison source.

Attribut de connexion supplémentaire	Valeur	Remarques
`applicationIntent`	`ReadOnly`	Sans ce paramètre ODBC, la tâche de réplication est routée vers le réplica du groupe de disponibilité principal. Pour plus d'informations, consultez Prise en charge des fonctionnalités de récupération d'urgence, haute disponibilité par SQL Server Native Client dans la documentation de SQL Server.
`multiSubnetFailover`	`yes`	Pour plus d'informations, consultez Prise en charge des fonctionnalités de récupération d'urgence, haute disponibilité par SQL Server Native Client dans la documentation de SQL Server.
`alwaysOnSharedSyncedBackupIsEnabled`	`false`	Pour de plus amples informations, veuillez consulter Paramètres du point de terminaison lors de l'utilisation de SQL Server comme source pour AWS DMS.
`activateSafeguard`	`false`	Pour plus d'informations, consultez Limites, ci-après.
`setUpMsCdcForTables`	`false`	Pour plus d'informations, consultez Limites, ci-après.

5. Activez l'option de distribution sur tous les réplicas de votre groupe de disponibilité. Ajoutez tous les nœuds à la liste des distributeurs. Pour de plus amples informations, veuillez consulter Pour configurer la distribution.
6. Exécutez la requête suivante sur le réplica principal en lecture-écriture pour activer la publication de la base de données. Vous n'exécutez cette requête qu'une seule fois pour la base de données.

```
sp_replicationdboption @dbname = N'<source DB name>', @optname = N'publish', @value
 = N'true';
```

Limites

Les limitations de l'utilisation d'un réplica de groupe de disponibilité secondaire sont les suivantes :

- AWS DMS ne prend pas en charge Safeguard lors de l'utilisation d'une réplique de groupe de disponibilité en lecture seule comme source. Pour de plus amples informations, veuillez consulter Paramètres du point de terminaison lors de l'utilisation de SQL Server comme source pour AWS DMS.
- AWS DMS ne prend pas en charge l'attribut de connexion `setUpMsCdcForTables` supplémentaire lors de l'utilisation d'une réplique de groupe de disponibilité en lecture seule comme source. Pour de plus amples informations, veuillez consulter Paramètres du point de terminaison lors de l'utilisation de SQL Server comme source pour AWS DMS.
- AWS DMS peut utiliser une réplique de groupe de disponibilité secondaire autogérée comme base de données source pour une réplication continue (capture des données de modification, ou CDC) à partir de la version 3.4.7. Les réplicas de lecture SQL Server Multi-AZ cloud ne sont pas pris en charge. Si vous utilisez des versions précédentes de AWS DMS, assurez-vous d'utiliser la réplique du groupe de disponibilité principal comme base de données source pour CDC.

Basculement vers d'autres nœuds

Si vous définissez l'attribut de connexion `ApplicationIntent` supplémentaire pour votre point de terminaison sur`ReadOnly`, votre AWS DMS tâche se connecte au nœud en lecture seule ayant la priorité de routage en lecture seule la plus élevée. Il bascule ensuite vers les autres nœuds en lecture seule de votre groupe de disponibilité lorsque le nœud en lecture seule ayant la priorité la plus élevée n'est pas disponible. Si vous ne le définissez pas`ApplicationIntent`, votre AWS DMS tâche se connecte uniquement au nœud principal (lecture/écriture) de votre groupe de disponibilité.

Exigences de sécurité lors de l'utilisation de SQL Server comme source pour AWS Database Migration Service

Le compte AWS DMS utilisateur doit avoir au moins le rôle db_owner d'utilisateur dans la base de données SQL Server source à laquelle vous vous connectez.

Paramètres du point de terminaison lors de l'utilisation de SQL Server comme source pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données source SQL Server comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison source à l'aide de la AWS DMS console ou à l'aide de la `create-endpoint` commande dans le AWS CLI, avec la syntaxe `--microsoft-sql-server-settings '{"EndpointSetting": "value", ...}'` JSON.

Les paramètres de point de terminaison que vous pouvez utiliser avec SQL Server en tant que source sont indiqués dans le tableau suivant.

Name (Nom)	Description
`ActivateSafeguard`	Cet attribut active ou désactive Safeguard. Pour en savoir plus sur Safeguard, consultez `SafeguardPolicy` ci-dessous. Valeur par défaut : `true` Valeurs valides : {`false`, `true`} Exemple : `'{"ActivateSafeguard": true}'`
`AlwaysOnSharedSynchedBackupIsEnabled`	Cet attribut ajuste le comportement AWS DMS lors de la migration depuis une base de données source SQL Server hébergée dans le cadre d'un cluster de groupes de disponibilité Always On. AWS DMS offre une prise en charge améliorée des bases de données source SQL Server configurées pour s'exécuter dans un cluster Always On. Dans ce cas, AWS DMS tente de savoir si des sauvegardes de transactions

Name (Nom)	Description
	sont effectuées à partir de nœuds du cluster Always On autres que le nœud sur lequel l'instance de base de données source est hébergée. Au démarrage de la tâche de migration, AWS DMS essaie de se connecter à chaque nœud du cluster, mais échoue s'il ne parvient pas à se connecter à l'un des nœuds. Si vous AWS DMS devez interroger tous les nœuds du cluster Always On pour les sauvegardes de transactions, définissez cet attribut sur`false`. Valeur par défaut : `true` Valeurs valides : `true` ou `false` Exemple : `'{"AlwaysOnSharedSynchedBackupIsEnabled": false}'`
`"ApplicationIntent": "readonly"`	Ce paramètre d'attribut du pilote ODBC oblige SQL Server à router votre tâche de réplication vers le nœud en lecture seule ayant la priorité la plus élevée. Sans ce paramètre, SQL Server route votre tâche de réplication vers le nœud en lecture-écriture principal.

Name (Nom)	Description
`EnableNonSysadminWrapper`	Utilisez ce paramètre de point de terminaison lorsque vous configurez la réplication continue sur un serveur SQL Server autonome sans utilisateur sysadmin. Ce paramètre est pris en charge dans les AWS DMS versions 3.4.7 et supérieures. Pour en savoir plus sur la configuration de la réplication continue sur une instance SQL Server autonome, consultez Configuration de la réplication continue sur une instance SQL Server autonome : sans le rôle sysadmin. Valeur par défaut : `false` Valeurs valides : `true`, `false` Exemple : `'{"EnableNonSysadminWrapper": true}'`
`ExecuteTimeout`	Utilisez cet attribut de connexion supplémentaire (ECA) pour définir le délai d'expiration de l'instruction client pour l'instance SQL Server, en secondes. La valeur par défaut est de 60 secondes. Exemple : `'{"ExecuteTimeout": 100}'`
`FatalOnSimpleModel`	Lorsqu'il est défini sur `true`, ce paramètre génère une erreur fatale lorsque le modèle de récupération de base de données SQL Server est défini sur `simple`. Valeur par défaut : `false` Valeurs valides : `true` ou `false` Exemple : `'{"FatalOnSimpleModel": true}'`

Name (Nom)	Description
`ForceLobLookup`	Force la recherche d'objets LOB sur le LOB en ligne. Valeur par défaut : `false` Valeurs valides : `true`, `false` Exemple : `'{"ForceLobLookup": false}'`
`"MultiSubnetFailover": "Yes"`	Cet attribut de pilote ODBC permet à DMS de se connecter au nouveau groupe de disponibilité principal en cas de basculement d'un groupe de disponibilité. Cet attribut a été conçu pour les situations où la connexion est rompue ou si l'adresse IP de l'écouteur est incorrecte. Dans ces situations, AWS DMS tente de se connecter à toutes les adresses IP associées à l'écouteur Availability Group.
`ReadBackupOnly`	L'utilisation de cet attribut nécessite des privilèges sysadmin. Lorsque cet attribut est défini surY, pendant la réplication en cours, AWS DMS lit les modifications uniquement à partir des sauvegardes du journal des transactions et non à partir du fichier journal des transacti ons actif. Définir ce paramètre sur `Y` vous permet de contrôler la croissance du fichier journal de transactions actives durant les tâches de chargement complet et de réplication continue. Toutefois, cela peut ajouter une latence source pour la réplication en cours. Valeurs valides : `N` ou `Y`. L'argument par défaut est `N`. Exemple : `'{"ReadBackupOnly": Y}'` Remarque : ce paramètre ne fonctionne pas sur les instances sources Amazon RDS SQL Server en raison de la manière dont RDS effectue les sauvegardes.

Name (Nom)	Description
`SafeguardPolicy`	Pour des performances optimales, AWS DMS essaie de capturer toutes les modifications non lues dans le journal des transactions actif (TLOG). Toutefois, il arrive parfois qu'en raison d'une troncation, le TLOG actif ne contienne pas toutes les modifications non lues. Lorsque cela se produit, AWS DMS accède à la sauvegarde du journal pour capturer les modifications manquantes. Pour minimiser le besoin d'accéder à la sauvegarde du journal, AWS DMS empêchez la troncature à l'aide de l'une des méthodes suivantes : 1. `RELY_ON_SQL_SERVER_REPLICATION_AGENT` (Lancer les transactions dans la base de données) : il s'agit de la valeur par défaut pour AWS DMS. Lorsque vous utilisez ce paramètre, AWS DMS requiert que l'agent de lecture de journaux SQL Server soit en cours d'exécution, afin qu' AWS DMS puisse déplacer les transactions marquées pour réplication à partir du TLOG actif. Notez que si l'agent de lecture de journaux n'est pas en cours d'exécution, le TLOG actif peut se remplir, ce qui bascule la base de données source en mode lecture seule jusqu'à ce que vous puissiez résoudre le problème. Si vous devez activer Microsoft Replication dans votre base de données dans un autre but AWS DMS, vous devez choisir ce paramètre. Lorsque vous utilisez ce paramètre, il AWS DMS minimise les lectures de sauvegarde du journal en créant une table appelée `awsdms_truncation_safeguard` et empêche la troncature du TLOG en imitant une transaction ouverte dans la base de données. Cela empêche la base de données de tronquer les événements et de les déplacer vers le journal de sauvegarde pendant cinq minutes (par

Name (Nom)	Description
	défaut). Assurez-vous que la table n'est incluse dans aucun plan de maintenance, car cela pourrait entraîner l'échec de la tâche de maintenance. Vous pouvez supprimer la table en toute sécurité si aucune tâche n'est configurée avec l'option de base de données `Start Transactions` . 2. `EXCLUSIVE_AUTOMATIC_TRUNCATION` (À utiliser exclusivement **`sp_repldone`** avec une seule tâche) : lorsque vous utilisez ce paramètre , AWS DMS il contrôle totalement le processus de l'agent de réplication qui marque les entrées du journal comme étant `ready for truncation` utilisées `sp_repldone` . Avec ce paramètre, AWS DMS n'utilise pas de transaction fictive comme avec le paramètre `RELY_ON_SQL_SERVER_REPLICATION_AGENT` (par défaut). Vous ne pouvez utiliser ce paramètre que lorsque MS Replication n'est pas utilisé à d'autres fins que dans la AWS DMS base de données source. En outre, lorsque vous utilisez ce paramètre, une seule AWS DMS tâche peut accéder à la base de données. Si vous devez exécuter des AWS DMS tâches parallèles sur la même base de données, utilisez`RELY_ON_SQL_SERVER_REPLICATION_AGENT` . • Ce paramètre nécessite que l'agent de lecture de journaux soit arrêté dans la base de données. Si l'agent Log Reader est en cours d'exécution au démarrage de la AWS DMS tâche, celle-ci forcera son arrêt. Vous pouvez également arrêter l'agent de lecture de journaux manuellement avant de démarrer la tâche. • Lorsque vous utilisez cette méthode avec MS-CDC, vous devez arrêter et désactiver les tâches de capture MS-CDC et de nettoyage MS-CDC.

Name (Nom)	Description
	• Vous ne pouvez pas utiliser ce paramètre lorsque la tâche de migration de Microsoft SQL Server s'exécute sur une machine de distribution distante, car elle AWS DMS n'a pas accès à la machine distante. • `EXCLUSIVE_AUTOMATIC_TRUNCATION` ne fonctionne pas sur les instances sources Amazon RDS for SQL Server, car les utilisateurs d'Amazon RDS ne sont pas autorisés à exécuter la procédure stockée `sp_repldone` . • Si vous définissez `SafeguardPolicy` sur `EXCLUSIVE_AUTOMATIC_TRUNCATION` sans utiliser le rôle sysadmin, vous devez accorder des autorisations sur les objets `dbo.syscategories` et `dbo.sysjobs` à l'utilisateur `dmsuser`. Valeur par défaut : `RELY_ON_SQL_SERVER_REPLICATION_AGENT` Valeurs valides : {`EXCLUSIVE_AUTOMATIC_TRUNCATION` , `RELY_ON_SQL_SERVER_REPLICATION_AGENT` } Exemple : `'{"SafeguardPolicy": "EXCLUSIVE_AUTOMATIC_TRUNCATION"}'`

Name (Nom)	Description
`SetUpMsCdcForTables`	Cet attribut active MS-CDC pour la base de données source et pour les tables du mappage de tâche pour lesquelles la réplication Microsoft n'a pas été activée. La définition de cette valeur sur `true` exécute la procédure stockée `sp_cdc_enable_db` sur la base de données source et exécute la procédure stockée `sp_cdc_enable_table` sur chaque table de la tâche pour laquelle la réplication Microsoft n'est pas activée dans la base de données source. Pour plus d'informations sur l'activation de la distribution, consultez Configuration de la réplication continue sur une instance SQL Server autogérée. Valeurs valides : {`true`, `false`} Exemple : `'{"SetUpMsCdcForTables": true}'`
`TlogAccessMode`	Indique le mode utilisé pour extraire les données de CDC. Valeur par défaut : `PreferTlog` Valeurs valides : `BackupOnly` , `PreferBackup` , `PreferTlog` , `TlogOnly` Exemple : `'{"TlogAccessMode": "PreferTlog"}'`
`Use3rdPartyBackupDevice`	Lorsque cet attribut est défini sur `Y`, AWS DMS traite les sauvegardes des journaux de transactions tiers si elles sont créées au format natif.

Types de données sources pour SQL Server

La migration de données qui utilise SQL Server comme source AWS DMS prend en charge la plupart des types de données SQL Server. Le tableau suivant indique les types de données source SQL

Server pris en charge lors de l'utilisation AWS DMS et le mappage par défaut à partir AWS DMS des types de données.

Pour en savoir plus sur la façon d'afficher le type de données qui est mappé dans la cible, consultez la section concernant le point de terminaison cible que vous utilisez.

Pour plus d'informations sur AWS DMS les types de données, consultezTypes de données pour AWS Database Migration Service.

Types de données SQL Server	AWS DMS types de données
BIGINT	INT8
BIT	BOOLEAN
DECIMAL	NUMERIC
INT	INT4
MONEY	NUMERIC
NUMERIC (p,s)	NUMERIC
SMALLINT	INT2
SMALLMONEY	NUMERIC
TINYINT	UINT1
REAL	REAL4
FLOAT	REAL8
DATETIME	DATETIME
DATETIME2 (SQL Server 2008 et versions ultérieures)	DATETIME
SMALLDATETIME	DATETIME
DATE	DATE

Types de données SQL Server	AWS DMS types de données
TIME	TIME
DATETIMEOFFSET	WSTRING
CHAR	CHAÎNE
VARCHAR	CHAÎNE
VARCHAR (max)	CLOB TEXT Pour utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation des types de données CLOB pour une tâche spécifique. Pour les tables SQL Server, AWS DMS met à jour les colonnes LOB de la cible, même pour les instructions UPDATE qui ne modifient pas la valeur de la colonne LOB dans SQL Server. Pendant le CDC, AWS DMS prend en charge les types de données CLOB uniquement dans les tables qui incluent une clé primaire.
NCHAR	WSTRING
NVARCHAR (length)	WSTRING

Types de données SQL Server	AWS DMS types de données
NVARCHAR (max)	NCLOB NTEXT Pour utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation de SupportLobs pour une tâche spécifique. Pour plus d'informations sur l'activation de la prise en charge des objets LOB, consultez Configuration du support LOB pour les bases de données sources dans une tâche AWS DMS. Pour les tables SQL Server, AWS DMS met à jour les colonnes LOB de la cible, même pour les instructions UPDATE qui ne modifient pas la valeur de la colonne LOB dans SQL Server. Pendant le CDC, AWS DMS prend en charge les types de données CLOB uniquement dans les tables qui incluent une clé primaire.
BINAIRE	BYTES
VARBINARY	BYTES

Types de données SQL Server	AWS DMS types de données
VARBINARY (max)	BLOB IMAGE Pour les tables SQL Server, AWS DMS met à jour les colonnes LOB de la cible, même pour les instructions UPDATE qui ne modifient pas la valeur de la colonne LOB dans SQL Server. Pour utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation des types de données BLOB pour une tâche spécifique. AWS DMS prend en charge les types de données BLOB uniquement dans les tables qui incluent une clé primaire.
TIMESTAMP	BYTES
UNIQUEIDENTIFIER	CHAÎNE
HIERARCHYID	Utilisez HIERARCHYID lors de la réplication sur un point de terminaison cible SQL Server. Utilisez WSTRING (250) lors de la réplication sur tous les autres points de terminaison cibles.

Types de données SQL Server	AWS DMS types de données
xml	NCLOB Pour les tables SQL Server, AWS DMS met à jour les colonnes LOB de la cible, même pour les instructions UPDATE qui ne modifient pas la valeur de la colonne LOB dans SQL Server. Pour utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation des types de données NCLOB pour une tâche spécifique. Pendant le CDC, AWS DMS prend en charge les types de données NCLOB uniquement dans les tables qui incluent une clé primaire.
GEOMETRY	Utilisez GEOMETRY lors de la réplication de points de terminaison cibles qui prennent en charge ce type de données. Utilisez CLOB lors de la réplication de points de terminaison cibles qui ne prennent pas en charge ce type de données.
GEOGRAPHY	Utilisez GEOGRAPHY lors de la réplication de points de terminaison cibles qui prennent en charge ce type de données. Utilisez CLOB lors de la réplication de points de terminaison cibles qui ne prennent pas en charge ce type de données.

AWS DMS ne prend pas en charge les tables qui incluent des champs contenant les types de données suivants.

- CURSOR
- SQL_VARIANT

- TABLE

> **Note**
>
> Les types de données définis par l'utilisateur sont pris en charge selon leur type de base. Par exemple, un type de données défini par l'utilisateur basé sur DATETIME est traité en tant que type de données DATETIME.

Utilisation de Microsoft Azure SQL Database comme source pour AWS DMS

Avec AWS DMS, vous pouvez utiliser Microsoft Azure SQL Database en tant que source de la même façon que vous le faites avec SQL Server. AWS DMS prend en charge, en tant que source, la même liste de versions de base de données que celles prises en charge pour SQL Server qui s'exécutent sur site ou sur une instance Amazon EC2.

Pour de plus amples informations, veuillez consulter Utilisation d'une base de données Microsoft SQL Server comme source pour AWS DMS.

> **Note**
>
> AWS DMS ne prend pas en charge les opérations de capture des données modifiées (CDC) avec Azure SQL Database.

Utilisation de Microsoft Azure SQL Managed Instance en tant que source pour AWS DMS

Avec AWS DMS, vous pouvez utiliser Microsoft Azure SQL Managed Instance en tant que source de la même façon que vous le faites avec SQL Server. AWS DMS prend en charge, en tant que source, la même liste de versions de base de données que celles prises en charge pour SQL Server qui s'exécutent sur site ou sur une instance Amazon EC2..

Pour de plus amples informations, veuillez consulter Utilisation d'une base de données Microsoft SQL Server comme source pour AWS DMS.

Utilisation du serveur flexible Microsoft Azure Database pour PostgreSQL en tant que source pour AWS DMS

Avec AWS DMS, vous pouvez utiliser le serveur flexible Microsoft Azure SQL Database pour PostgreSQL en tant que source de la même façon que vous le faites avec PostgreSQL.

Pour en savoir plus sur les versions du serveur flexible Microsoft Azure Database pour PostgreSQL qui sont prises en charge par AWS DMS en tant que source, consultez Sources pour AWS DMS.

Configuration du serveur flexible Microsoft Azure pour PostgreSQL pour la réplication logique et le décodage

Vous pouvez utiliser les fonctionnalités de réplication logique et de décodage du serveur flexible Microsoft Azure Database pour PostgreSQL dans le cadre de la migration de base de données.

Pour le décodage logique, DMS utilise le plug-in `test_decoding` ou `pglogical`. Si le plug-in `pglogical` est disponible sur une base de données PostgreSQL source, DMS crée un emplacement de réplication en utilisant `pglogical`, sinon le plug-in `test_decoding` est utilisé.

Pour configurer votre serveur flexible Microsoft Azure pour PostgreSQL en tant que point de terminaison source pour DMS, effectuez les étapes suivantes :

1. Ouvrez la page Paramètres du serveur sur le portail.
2. Définissez le paramètre `wal_level` du serveur sur `LOGICAL`.
3. Si vous souhaitez utiliser l'extension `pglogical`, définissez les paramètres `shared_preload_libraries` et `azure.extensions` sur `pglogical`.
4. Définissez le paramètre `max_replication_slots` sur le nombre maximal de tâches DMS que vous envisagez d'exécuter simultanément. Dans Microsoft Azure, la valeur par défaut de ce paramètre est 10. La valeur maximale de ce paramètre dépend de la mémoire disponible de votre instance PostgreSQL, ce qui permet de prévoir entre 2 et 8 emplacements de réplication par Go de mémoire.
5. Définissez le paramètre `max_wal_senders` sur une valeur supérieure à 1. Le paramètre `max_wal_senders`définit le nombre de tâches simultanées qui peuvent s'exécuter. La valeur par défaut est 10.
6. Définissez la valeur du paramètre `max_worker_processes` sur au moins 16. Dans le cas contraire, des erreurs telles que les suivantes peuvent survenir :

```
WARNING: out of background worker slots.
```

7. Enregistrez les modifications. Redémarrez le serveur pour appliquer les modifications.
8. Vérifiez que votre instance PostgreSQL autorise le trafic réseau provenant de votre ressource de connexion.
9. Accordez des autorisations de réplication à un utilisateur existant ou créez un nouvel utilisateur avec des autorisations de réplication à l'aide des commandes suivantes.

   - Accordez des autorisations de réplication à un utilisateur existant à l'aide de la commande suivante :

     ```
     ALTER USER <existing_user> WITH REPLICATION;
     ```

   - Créez un nouvel utilisateur avec des autorisations de réplication à l'aide de la commande suivante :

     ```
     CREATE USER aws_dms_user PASSWORD 'aws_dms_user_password';
     GRANT azure_pg_admin to aws_dms_user;
     ALTER ROLE aws_dms_user REPLICATION LOGIN;
     ```

Pour plus d'informations sur la réplication logique avec PostgreSQL, consultez les rubriques suivantes :

- Activation de la capture des données de modification (CDC) à l'aide de la réplication logique
- Utilisation de points de départ CDC natifs pour configurer la charge CDC d'une source PostgreSQL
- Réplication logique et décodage logique dans le serveur flexible Azure Database pour PostgreSQL dans la documentation Azure Database pour PostgreSQL.

Utilisation du serveur flexible Microsoft Azure Database pour MySQL en tant que source pour AWS DMS

Avec AWS DMS, vous pouvez utiliser le serveur flexible Microsoft Azure Database pour MySQL en tant que source de la même manière que MySQL.

Pour en savoir plus sur les versions du serveur flexible Microsoft Azure Database pour MySQL qui sont prises en charge par AWS DMS en tant que source, consultez Sources pour AWS DMS.

Pour plus d'informations sur l'utilisation d'une base de données compatible MySQL gérée par le client avec AWS DMS, consultez Utilisation d'une base de données compatible MySQL autogérée comme source pour AWS DMS.

Limitations relatives à l'utilisation d'Azure MySQL en tant que source pour AWS Database Migration Service

- La valeur par défaut de la variable système du serveur flexible Azure MySQL `sql_generate_invisible_primary_key` est `ON`, et le serveur ajoute automatiquement une clé primaire invisible générée (GIPK) à toute table créée sans clé primaire explicite. AWS DMS ne prend pas en charge la réplication continue pour les tables MySQL avec des contraintes GIPK.

Utilisation d'OCI MySQL Heatwave en tant que source pour AWS DMS

Avec AWS DMS, vous pouvez utiliser OCI MySQL Heatwave en tant que source de la même manière que MySQL. L'utilisation d'OCI MySQL Heatwave en tant que source nécessite quelques modifications de configuration supplémentaires.

Pour en savoir plus sur les versions d'OCI MySQL Heatwave qu'AWS DMS prend en charge en tant que source, consultez Sources pour AWS DMS.

Configuration d'OCI MySQL Heatwave pour la réplication logique

Pour configurer votre instance OCI MySQL Heatwave en tant que point de terminaison source pour DMS, procédez comme suit :

1. Connectez-vous à la console OCI et ouvrez le menu principal des hamburgers (≡) dans le coin supérieur gauche.
2. Choisissez Bases de données, Systèmes de bases de données.
3. Ouvrez le menu Configurations.
4. Choisissez Create configuration (Créer une configuration).
5. Entrez un nom de configuration, tel que **`dms_configuration`**.
6. Choisissez la forme de votre instance OCI MySQL Heatwave actuelle. Vous pouvez trouver la forme dans l'onglet de propriétés Configuration du système de bases de données de l'instance, dans la section Configuration du système de base de données : Forme.
7. Dans la section Variables d'utilisateur, choisissez la variable système `binlog_row_value_options`. Sa valeur par défaut est `PARTIAL_JSON`. Effacez la valeur.

8. Choisissez le bouton Créer.
9. Ouvrez votre instance OCI MySQL Heatwave et choisissez le bouton Modifier.
10. Dans la section Configuration, choisissez le bouton Modifier la configuration, puis choisissez la configuration de forme que vous avez créée à l'étape 4.
11. Une fois que les modifications ont pris effet, votre instance est prête pour la réplication logique.

Utilisation de Google Cloud pour MySQL en tant que source pour AWS DMS

Avec AWS DMS, vous pouvez utiliser Google Cloud pour MySQL en tant que source de la même manière que MySQL.

Pour en savoir plus sur les versions de GSP MySQL prises en charge par AWS DMS en tant que source, consultez Sources pour AWS DMS.

Pour de plus amples informations, veuillez consulter Utilisation d'une base de données compatible MySQL comme source pour AWS DMS.

> **Note**
>
> La prise en charge de GCP MySQL 8.0 en tant que source est disponible dans AWS DMS version 3.4.6.
> AWS DMS ne prend pas en charge le mode SSL `verify-full` pour les instances GCP pour MySQL.
> Le paramètre de sécurité GCP MySQL `Allow only SSL connections` n'est pas pris en charge, car il nécessite la vérification des certificats de serveur et client. AWS DMS prend uniquement en charge la vérification des certificats de serveur.
> AWS DMS prend en charge la valeur par défaut de GCP CloudSQL pour MySQL `CRC32` pour l'indicateur de base de données `binlog_checksum`.

Utilisation de Google Cloud pour PostgreSQL en tant que source pour AWS DMS

Avec AWS DMS, vous pouvez utiliser Google Cloud pour PostgreSQL en tant que source de la même manière que les bases de données PostgreSQL autogérées.

Pour en savoir plus sur les versions de GCP PostgreSQL prises en charge par AWS DMS en tant que source, consultez Sources pour AWS DMS.

Pour de plus amples informations, veuillez consulter Utilisation d'une base de données PostgreSQL en tant que source AWS DMS.

Configuration de Google Cloud pour PostgreSQL pour la réplication logique et le décodage

Vous pouvez utiliser les fonctionnalités de réplication logique et de décodage dans Google Cloud SQL pour PostgreSQL au cours de la migration de base de données.

Pour le décodage logique, DMS utilise l'un des plug-ins suivants :

- `test_decoding`
- `pglogical`

Si le plug-in `pglogical` est disponible sur une base de données PostgreSQL source, DMS crée un emplacement de réplication en utilisant `pglogical`, sinon le plug-in `test_decoding` est utilisé.

Notez ce qui suit à propos de l'utilisation du décodage logique avec AWS DMS :

1. Avec Google Cloud SQL pour PostgreSQL, activez le décodage logique en attribuant à l'indicateur `cloudsql.logical_decoding` la valeur on.
2. Pour activer `pglogical`, attribuez à l'indicateur `cloudsql.enable_pglogical` la valeur on, puis redémarrez la base de données.
3. Pour utiliser les fonctionnalités de décodage logique, vous devez créer un utilisateur PostgreSQL avec l'attribut `REPLICATION`. Lorsque vous utilisez l'extension `pglogical`, l'utilisateur doit avoir le rôle `cloudsqlsuperuser`. Pour créer un utilisateur avec le rôle `cloudsqlsuperuser`, procédez comme suit :

   ```
   CREATE USER new_aws_dms_user WITH REPLICATION
   IN ROLE cloudsqlsuperuser LOGIN PASSWORD 'new_aws_dms_user_password';
   ```

   Pour définir cet attribut sur un utilisateur existant, procédez comme suit :

   ```
   ALTER USER existing_user WITH REPLICATION;
   ```

4. Définissez le paramètre `max_replication_slots` sur le nombre maximal de tâches DMS que vous envisagez d'exécuter simultanément. Dans Google Cloud SQL, la valeur par défaut de ce paramètre est 10. La valeur maximale de ce paramètre dépend de la mémoire disponible de votre instance PostgreSQL, ce qui permet de prévoir entre 2 et 8 emplacements de réplication par Go de mémoire.

Pour plus d'informations sur la réplication logique avec PostgreSQL, consultez les rubriques suivantes :

- Activation de la capture des données de modification (CDC) à l'aide de la réplication logique
- Utilisation de points de départ CDC natifs pour configurer la charge CDC d'une source PostgreSQL
- Configurer la réplication logique et le décodage dans la documentation de Cloud SQL pour PostgreSQL.

Utilisation d'une base de données PostgreSQL en tant que source AWS DMS

Vous pouvez migrer des données d'une ou plusieurs bases de données PostgreSQL à l'aide d'AWS DMS. Avec une base de données PostgreSQL en tant que source, vous pouvez migrer les données vers une autre base de données PostgreSQL ou vers l'une des autres bases de données prises en charge.

Pour en savoir plus sur les versions de PostgreSQL qui sont prises en charge par AWS DMS en tant que source, consultez Sources pour AWS DMS.

AWS DMS prend en charge PostgreSQL pour les types de bases de données suivants :

- Bases de données sur site
- Bases de données sur une instance Amazon EC2
- Bases de données sur une instance de base de données Amazon RDS
- Bases de données sur une instance de base de données basée sur Amazon Aurora PostgreSQL-Compatible Edition
- Bases de données sur une instance de base de données basée sur Amazon Aurora PostgreSQL-Compatible Edition sans serveur

> **Note**
>
> DMS prend en charge Amazon Aurora PostgreSQL sans serveur V1 en tant que source pour le chargement complet uniquement. Mais vous pouvez utiliser Amazon Aurora PostgreSQL sans serveur V2 en tant que source pour les tâches Chargement complet, Chargement complet + CDC et CDC uniquement.

Version AWS DMS à utiliser
Utilisez n'importe quelle version AWS DMS disponible.
Utilisez AWS DMS versions 3.4.3 et ultérieures.
Utilisez AWS DMS versions 3.4.7 et ultérieures.
Utilisez AWS DMS versions 3.5.1 et ultérieures.

Vous pouvez utiliser Secure Socket Layers (SSL) pour chiffrer les connexions entre votre point de terminaison PostgreSQL et l'instance de réplication. Pour plus d'informations sur l'utilisation de SSL avec un point de terminaison PostgreSQL, consultez Utilisation du protocole SSL avec AWS Database Migration Service.

Lorsque vous utilisez PostgreSQL en tant que source, la seule exigence de sécurité supplémentaire consiste à ce que le compte d'utilisateur spécifié soit un utilisateur enregistré dans la base de données PostgreSQL.

Pour configurer une base de données PostgreSQL en tant que point de terminaison source AWS DMS, procédez comme suit :

- Créez un utilisateur PostgreSQL doté des autorisations appropriées permettant à AWS DMS d'accéder à la base de données source PostgreSQL.

  > **Note**
  >
  > - Si la base de données source PostgreSQL est autogérée, consultez Utilisation de bases de données PostgreSQL autogérées en tant que source dans AWS DMS pour plus d'informations.
  > - Si la base de données source PostgreSQL est gérée par Amazon RDS, consultez Utilisation de bases de données PostgreSQL gérées par AWS en tant que source DMS pour plus d'informations.

- Créez un point de terminaison source PostgreSQL conforme à la configuration de base de données PostgreSQL que vous avez choisie.
- Créez une tâche ou un ensemble de tâches pour migrer vos tables.

  Pour créer une full-load-only tâche, aucune autre configuration de point de terminaison n'est nécessaire.

  Avant de créer une tâche pour la capture des données de modification (une tâche CDC uniquement ou une tâche de chargement complet + CDC), consultez Activation de la CDC à l'aide d'une base de données PostgreSQL autogérée en tant que source AWS DMS ou Activation de la CDC avec une instance de base de données PostgreSQL gérée par AWS avec AWS DMS.

Rubriques

- Utilisation de bases de données PostgreSQL autogérées en tant que source dans AWS DMS
- Utilisation de bases de données PostgreSQL gérées par AWS en tant que source DMS
- Activation de la capture des données de modification (CDC) à l'aide de la réplication logique
- Utilisation de points de départ CDC natifs pour configurer la charge CDC d'une source PostgreSQL
- Migration de PostgreSQL vers PostgreSQL à l'aide d'AWS DMS.
- Suppression d'objets AWS DMS d'une base de données source PostgreSQL.
- Paramètres de configuration supplémentaires lors de l'utilisation d'une base de données PostgreSQL en tant que source DMS
- Utilisation du paramètre de point de MapBooleanAsBoolean terminaison PostgreSQL

- Paramètres du point de terminaison et attributs de connexion supplémentaires (ECA) lors de l'utilisation de PostgreSQL comme source DMS
- Limitations de l'utilisation d'une base de données PostgreSQL en tant que source DMS
- Types de données sources pour PostgreSQL

Utilisation de bases de données PostgreSQL autogérées en tant que source dans AWS DMS

Avec une base de données PostgreSQL autogérée en tant que source, vous pouvez migrer les données vers une autre base de données PostgreSQL ou vers l'une des autres bases de données cibles prises en charge par AWS DMS. La base de données source peut être sur une base de données sur site ou un moteur autogéré s'exécutant sur une instance Amazon EC2. Vous pouvez utiliser une instance de base de données pour les tâches de chargement complet et pour les tâches de capture des données de modification (CDC).

Conditions requises pour l'utilisation d'une base de données PostgreSQL autogérée en tant que source AWS DMS

Avant de migrer des données à partir d'une base de données source PostgreSQL autogérée, procédez comme suit :

- Assurez-vous d'utiliser une base de données PostgreSQL version 9.4.x ou ultérieure.
- Pour les tâches de chargement complet + CDC ou les tâches de CDC uniquement, accordez des autorisations de super-utilisateur pour le compte d'utilisateur spécifié pour la base de données source PostgreSQL. Le compte d'utilisateur a besoin d'autorisations de super-utilisateur pour accéder aux fonctions spécifiques à la réplication dans la source. Pour les tâches de chargement complet uniquement, le compte d'utilisateur a besoin d'autorisations SELECT sur les tables afin de les migrer.
- Ajoutez l'adresse IP du serveur de réplication AWS DMS dans le fichier de configuration `pg_hba.conf` et activez les connexions de réplication et de socket. Un exemple suit.

```
# Replication Instance
host all all 12.3.4.56/00 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
host replication dms 12.3.4.56/00 md5
```

Le fichier de configuration `pg_hba.conf` de PostgreSQL contrôle l'authentification du client. (HBA signifie authentification basée sur l'hôte.) Le fichier est traditionnellement stocké dans le répertoire de données du cluster de base de données.

- Si vous configurez une base de données en tant que source pour la réplication logique à l'aide d'AWS DMS, consultez Activation de la CDC à l'aide d'une base de données PostgreSQL autogérée en tant que source AWS DMS.

> **Note**
>
> Certaines transactions AWS DMS sont inactives un certain temps avant que le moteur DMS ne les utilise à nouveau. En utilisant le paramètre `idle_in_transaction_session_timeout` dans PostgreSQL versions 9.6 et ultérieures, vous pouvez provoquer l'expiration et l'échec des transactions inactives. Ne mettez pas fin aux transactions inactives lorsque vous utilisez AWS DMS.

Activation de la CDC à l'aide d'une base de données PostgreSQL autogérée en tant que source AWS DMS

AWS DMS prend en charge la capture des données de modification (CDC) à l'aide de la réplication logique. Pour activer la réplication logique d'une base de données source PostgreSQL autogérée, définissez les paramètres suivants dans le fichier de configuration `postgresql.conf` :

- Configurez `wal_level = logical`.
- Définissez une valeur supérieure à 1 pour `max_replication_slots`.

  Définissez la valeur `max_replication_slots` selon le nombre de tâches que vous souhaitez exécuter. Par exemple, pour exécuter cinq tâches, vous définissez au moins cinq emplacements. Les emplacements s'ouvrent automatiquement dès qu'une tâche commence et restent ouvert même lorsque la tâche n'est plus en cours d'exécution. Assurez-vous de supprimer manuellement les emplacements ouverts. Notez que DMS supprime automatiquement les emplacements de réplication lorsque la tâche est supprimée, si DMS les a créés.
- Définissez une valeur supérieure à 1 pour `max_wal_senders`.

  Le paramètre `max_wal_senders`définit le nombre de tâches simultanées qui peuvent s'exécuter.

- Le paramètre `wal_sender_timeout` met fin aux connexions de réplication qui sont inactives plus longtemps que le nombre de millisecondes spécifié. La valeur par défaut pour une base de données PostgreSQL sur site est de 60 000 millisecondes (60 secondes). La définition de la valeur sur 0 (zéro) désactive le mécanisme d'expiration et constitue une valeur valide pour DMS.

  Lorsque vous définissez `wal_sender_timeout` sur une valeur différente de zéro, une tâche DMS avec CDC nécessite au moins 10 000 millisecondes (10 secondes) et échoue si la valeur est inférieure à 10 000. Définissez une valeur inférieure à 5 minutes pour éviter tout retard lors du basculement multi-AZ d'une instance de réplication DMS.

Certains paramètres sont statiques et vous ne pouvez les définir qu'au démarrage du serveur. Toute modification apportée à leurs entrées dans le fichier de configuration (pour une base de données autogérée) ou le groupe de paramètres de base de données (pour une base de données RDS for PostgreSQL) est ignorée jusqu'au redémarrage du serveur. Pour de plus amples informations, veuillez consulter la documentation sur PostgreSQL.

Pour plus d'informations sur l'activation de la CDC, consultez Activation de la capture des données de modification (CDC) à l'aide de la réplication logique.

Utilisation de bases de données PostgreSQL gérées par AWS en tant que source DMS

Vous pouvez utiliser une instance de base de données PostgreSQL gérée par AWS en tant que source pour AWS DMS. Vous pouvez effectuer à la fois des tâches de chargement complet et des tâches de capture des données de modification (CDC) à l'aide d'une source PostgreSQL gérée par AWS.

Conditions préalables à l'utilisation d'une base de données PostgreSQL gérée par AWS en tant que source DMS

Avant de migrer des données à partir d'une base de données source PostgreSQL gérée par AWS, procédez comme suit :

- Nous vous recommandons d'utiliser un compte d'utilisateur AWS avec les autorisations minimales requises pour l'instance de base de données PostgreSQL en tant que compte d'utilisateur pour le point de terminaison source PostgreSQL pour AWS DMS. L'utilisation du compte principal n'est pas recommandée. Le compte doit avoir le rôle `rds_superuser` et le rôle `rds_replication`. Le rôle `rds_replication` accorde les autorisations permettant de gérer des emplacements logiques et de diffuser les données à l'aide d'emplacements logiques.

Assurez-vous de créer plusieurs objets à partir du compte d'utilisateur principal pour le compte que vous utilisez. Pour en savoir plus sur la création de ces objets, consultez Migration d'une base de données Amazon RDS for PostgreSQL sans utiliser le compte d'utilisateur principal.

- Si la base de données source se trouve dans un cloud privé virtuel (VPC), choisissez le groupe de sécurité de VPC qui permet d'accéder à l'instance de base de données où réside la base de données. Cette action est nécessaire pour que l'instance de réplication DMS se connecte avec succès à l'instance de base de données source. Lorsque la base de données et l'instance de réplication DMS se trouvent dans le même VPC, ajoutez le groupe de sécurité approprié à ses propres règles entrantes.

> **Note**
>
> Certaines transactions AWS DMS sont inactives un certain temps avant que le moteur DMS ne les utilise à nouveau. En utilisant le paramètre `idle_in_transaction_session_timeout` dans PostgreSQL versions 9.6 et ultérieures, vous pouvez provoquer l'expiration et l'échec des transactions inactives. Ne mettez pas fin aux transactions inactives lorsque vous utilisez AWS DMS.

Activation de la CDC avec une instance de base de données PostgreSQL gérée par AWS avec AWS DMS

AWS DMS prend en charge la CDC sur les bases de données Amazon RDS PostgreSQL lorsque l'instance de base de données est configurée pour utiliser la réplication logique. Le tableau suivant récapitule la compatibilité de la réplication logique avec chaque version de PostgreSQL gérée par AWS.

Vous ne pouvez pas utiliser les réplicas en lecture RDS PostgreSQL pour la CDC (réplication continue).

Version PostgreSQL	Prise en charge du chargement complet AWS DMS	Prise en charge de la CDC AWS DMS
Compatibilité d'Aurora PostgreSQL version 2.1 avec	Oui	Non

Version PostgreSQL	Prise en charge du chargement complet AWS DMS	Prise en charge de la CDC AWS DMS
PostgreSQL 10.5 (ou antérieur)		
Compatibilité d'Aurora PostgreSQL version 2.2 avec PostgreSQL 10.6 (ou ultérieur)	Oui	Oui
Compatibilité de RDS for PostgreSQL avec PostgreSQL 10.21 (ou ultérieur)	Oui	Oui

Pour activer la réplication logique pour une instance DB RDS PostgreSQL

1. Utilisez le compte d'utilisateur principal AWS pour l'instance de base de données PostgreSQL en tant que compte d'utilisateur pour le point de terminaison source PostgreSQL. Le compte d'utilisateur principal a les rôles nécessaires qui lui permettent de configurer la capture des données modifiées.

   Si vous utilisez un compte autre que le compte d'utilisateur principal, assurez-vous de créer plusieurs objets à partir du compte principal pour le compte que vous utilisez. Pour plus d'informations, consultez Migration d'une base de données Amazon RDS for PostgreSQL sans utiliser le compte d'utilisateur principal.
2. Définissez le paramètre `rds.logical_replication` de votre groupe de paramètres DB CLUSTER sur 1. Ce paramètre statique nécessite un redémarrage de l'instance DB pour son application. Pour l'application de ce paramètre, AWS DMS définit les paramètres `wal_level`, `max_wal_senders`, `max_replication_slots` et `max_connections`. Ces modifications de paramètres peuvent augmenter la génération WAL (Write ahead log) ; de sorte qu'il faut uniquement définir `rds.logical_replication` lorsque vous utilisez des emplacements de réplication logique.
3. Le paramètre `wal_sender_timeout` met fin aux connexions de réplication qui sont inactives plus longtemps que le nombre de millisecondes spécifié. La valeur par défaut pour une base de données PostgreSQL gérée par AWS est de 30 000 millisecondes (30 secondes). La définition de la valeur sur 0 (zéro) désactive le mécanisme d'expiration et constitue une valeur valide pour DMS.

Lorsque vous définissez `wal_sender_timeout` sur une valeur différente de zéro, une tâche DMS avec CDC nécessite au moins 10 000 millisecondes (10 secondes) et échoue si la valeur est comprise entre 0 et 10 000. Définissez une valeur inférieure à 5 minutes pour éviter tout retard lors du basculement multi-AZ d'une instance de réplication DMS.

4. Assurez-vous que la valeur du paramètre `max_worker_processes` dans le groupe de paramètres de votre cluster de bases de données est supérieure ou égale aux valeurs combinées totales de `max_logical_replication_workers`, `autovacuum_max_workers` et `max_parallel_workers`. Un nombre élevé de processus de travail en arrière-plan peut avoir un impact sur les charges de travail d'application sur les instances de petite taille. Vous devez donc surveiller les performances de la base de données si vous définissez `max_worker_processes` sur une valeur supérieure à la valeur par défaut.

Migration d'une base de données Amazon RDS for PostgreSQL sans utiliser le compte d'utilisateur principal

Dans certains cas, vous risquez de ne pas utiliser le compte d'utilisateur principal pour l'instance de base de données Amazon RDS PostgreSQL que vous utilisez en tant que source. Dans ce cas, créez plusieurs objets afin de capturer des événements DDL (data definition language). Vous créez ces objets dans le compte autre que le compte principal, puis vous créez un déclencheur dans le compte utilisateur principal.

> **Note**
>
> Si vous définissez le paramètre de point de terminaison `captureDDLs` sur `false` sur le point de terminaison source, vous n'avez pas besoin de créer la table et le déclencheur suivants sur la base de données source.

Utilisez la procédure suivante pour créer ces objets.

Pour créer des objets

1. Choisissez le schéma où les objets doivent être créés. Le schéma par défaut est `public`. Assurez-vous que le schéma existe et qu'il est accessible par le compte *`OtherThanMaster`*.
2. Connectez-vous à l'instance de base de données PostgreSQL en utilisant un compte d'utilisateur autre que le compte principal, à savoir le compte *`OtherThanMaster`* dans notre cas.

3. Créez la table `awsdms_ddl_audit` en exécutant la commande suivante, en remplaçant *`objects_schema`* dans le code suivant par le nom du schéma à utiliser.

```
CREATE TABLE objects_schema.awsdms_ddl_audit
(
  c_key    bigserial primary key,
  c_time   timestamp,    -- Informational
  c_user   varchar(64),  -- Informational: current_user
  c_txn    varchar(16),  -- Informational: current transaction
  c_tag    varchar(24),  -- Either 'CREATE TABLE' or 'ALTER TABLE' or 'DROP TABLE'
  c_oid    integer,      -- For future use - TG_OBJECTID
  c_name   varchar(64),  -- For future use - TG_OBJECTNAME
  c_schema varchar(64),  -- For future use - TG_SCHEMANAME. For now - holds
 current_schema
  c_ddlqry  text         -- The DDL query associated with the current DDL event
);
```

4. Créez la fonction `awsdms_intercept_ddl` en exécutant la commande suivante, en remplaçant *`objects_schema`* dans le code suivant par le nom du schéma à utiliser.

```
CREATE OR REPLACE FUNCTION objects_schema.awsdms_intercept_ddl()
  RETURNS event_trigger
LANGUAGE plpgsql
SECURITY DEFINER
  AS $$
  declare _qry text;
BEGIN
  if (tg_tag='CREATE TABLE' or tg_tag='ALTER TABLE' or tg_tag='DROP TABLE' or
 tg_tag = 'CREATE TABLE AS') then
         SELECT current_query() into _qry;
         insert into objects_schema.awsdms_ddl_audit
         values
         (
         default,current_timestamp,current_user,cast(TXID_CURRENT()as
 varchar(16)),tg_tag,0,'',current_schema,_qry
         );
         delete from objects_schema.awsdms_ddl_audit;
end if;
END;
```

```
$$;
```

5. Déconnectez-vous du compte *OtherThanMaster* et connectez-vous avec un compte auquel le rôle `rds_superuser` a été attribué.
6. Créez le déclencheur d'événements `awsdms_intercept_ddl` en exécutant la commande suivante.

```
CREATE EVENT TRIGGER awsdms_intercept_ddl ON ddl_command_end
EXECUTE PROCEDURE objects_schema.awsdms_intercept_ddl();
```

7. Assurez-vous que tous les utilisateurs et rôles qui accèdent à ces événements disposent des autorisations DDL nécessaires. Par exemple :

```
grant all on public.awsdms_ddl_audit to public;
grant all on public.awsdms_ddl_audit_c_key_seq to public;
```

Une fois que vous avez terminé la procédure précédente, vous pouvez créer le point de terminaison source AWS DMS à l'aide du compte *OtherThanMaster*.

> **Note**
>
> Ces événements sont déclenchés par les instructions `CREATE TABLE`, `ALTER TABLE` et `DROP TABLE`.

Activation de la capture des données de modification (CDC) à l'aide de la réplication logique

Vous pouvez utiliser la fonctionnalité de réplication logique native de PostgreSQL pour activer la capture des données de modification (CDC) lors de la migration de base de données pour les sources PostgreSQL. Vous pouvez utiliser cette fonctionnalité avec une instance de base de données PostgreSQL autogérée et avec une instance de base de données SQL Amazon RDS for PostgreSQL. Cette approche réduit les temps d'arrêt et permet de garantir que la base de données cible est synchronisée avec la base de données PostgreSQL source.

AWS DMS prend en charge la CDC pour les tables PostgreSQL avec des clés primaires. Si une table n'a pas de clé primaire, les journaux WAL n'incluent pas d'image antérieure de la ligne de base de données. Dans ce cas, DMS ne peut pas mettre à jour la table. Ici, vous pouvez utiliser des paramètres de configuration supplémentaires et l'identité du réplica de table comme solution de contournement. Toutefois, cette approche peut générer des journaux supplémentaires. Nous vous recommandons d'utiliser l'identité du réplica de table comme solution de contournement uniquement après avoir effectué des tests méticuleux. Pour plus d'informations, consultez Paramètres de configuration supplémentaires lors de l'utilisation d'une base de données PostgreSQL en tant que source DMS.

> **Note**
>
> REPLICA IDENTITY FULL est compatible avec un plug-in de décodage logique, mais pas avec un plug-in pglogical. Pour plus d'informations, consultez la documentation de pglogical.

Pour les tâches de chargement complet + CDC et les tâches de CDC uniquement, AWS DMS utilise des emplacements de réplication logique pour conserver les journaux WAL à des fins de réplication jusqu'à ce qu'ils soient décodés. Au redémarrage (et non à la reprise) pour une tâche de chargement complet + CDC ou une tâche de CDC, l'emplacement de réplication est recréé.

> **Note**
>
> Pour le décodage logique, DMS utilise le plug-in test_decoding ou pglogical. Si le plug-in pglogical est disponible sur une base de données PostgreSQL source, DMS crée un emplacement de réplication à l'aide de pglogical. Sinon, un plug-in test_decoding est utilisé. Pour plus d'informations sur le plug-in test_decoding, consultez la documentation de PostgreSQL.
>
> Si le paramètre de base de données `max_slot_wal_keep_size` est défini sur une valeur autre que celle par défaut et que le paramètre `restart_lsn` d'un emplacement de réplication dépasse la taille du LSN actuel, la tâche DMS échoue en raison de la suppression des fichiers WAL requis.

Configuration du plug-in pglogical

Mis en œuvre en tant qu'extension PostgreSQL, le plug-in pglogical est un système et un modèle de réplication logique pour la réplication sélective des données. Le tableau suivant identifie les versions de base de données PostgreSQL source qui prennent en charge le plug-in pglogical.

Source PostgreSQL	Prend en charge pglogical
PostgreSQL 9.4 autogéré ou version ultérieure	Oui
Amazon RDS PostgreSQL 9.5 ou version antérieure	Non
Amazon RDS PostgreSQL 9.6 ou version ultérieure	Oui
Aurora PostgreSQL 1.x jusqu'à 2.5.x	Non
Aurora PostgreSQL 2.6.x ou version ultérieure	Oui
Aurora PostgreSQL 3.3.x ou version ultérieure	Oui

Avant de configurer pglogical pour l'utiliser avec AWS DMS, commencez par activer la réplication logique pour la capture des données de modification (CDC) sur la base de données source PostgreSQL.

- Pour en savoir plus sur l'activation de la réplication logique pour la CDC sur les bases de données sources PostgreSQL autogérées, consultez Activation de la CDC à l'aide d'une base de données PostgreSQL autogérée en tant que source AWS DMS.
- Pour en savoir plus sur l'activation de la réplication logique pour la CDC sur les bases de données sources PostgreSQL gérées par AWS, consultez Activation de la CDC avec une instance de base de données PostgreSQL gérée par AWS avec AWS DMS.

Une fois la réplication logique activée sur la base de données source PostgreSQL, suivez les étapes ci-dessous pour configurer pglogical en vue de son utilisation avec DMS.

Pour utiliser le plug-in pglogical pour la réplication logique sur une base de données source PostgreSQL avec AWS DMS

1. Créez une extension pglogical sur la base de données PostgreSQL source :

   a. Définissez le paramètre correct :

      - Pour les bases de données PostgreSQL autogérées, définissez le paramètre de base de données `shared_preload_libraries= 'pglogical'`.
      - Pour les bases de données PostgreSQL sur Amazon RDS et Amazon Aurora PostgreSQL-Compatible Edition, définissez le paramètre `shared_preload_libraries` sur `pglogical` dans le même groupe de paramètres RDS.

   b. Redémarrez la base de données source PostgreSQL.

   c. Sur la base de données PostgreSQL, exécutez la commande `create extension pglogical;`.

2. Exécutez la commande suivante pour vérifier que pglogical a été correctement installé :

```
select * FROM pg_catalog.pg_extension
```

Vous pouvez désormais créer une tâche AWS DMS qui capture les données de modification pour le point de terminaison de la base de données source PostgreSQL.

> **Note**
>
> Si vous n'activez pas pglogical dans la base de données source PostgreSQL, AWS DMS utilise le plug-in `test_decoding` par défaut. Lorsque pglogical est activé pour le décodage logique, AWS DMS utilise pglogical par défaut. Mais vous pouvez définir l'attribut de connexion supplémentaire, `PluginName`, pour utiliser le plug-in `test_decoding` à la place.

Utilisation de points de départ CDC natifs pour configurer la charge CDC d'une source PostgreSQL

Pour activer les points de départ CDC natifs avec PostgreSQL en tant que source, définissez l'attribut de connexion supplémentaire `slotName` sur le nom d'un emplacement de réplication logique existant lorsque vous créez le point de terminaison. Cet emplacement de réplication logique contient les

modifications en cours à partir du moment de la création du point de terminaison, il prend donc en charge la réplication à partir d'un point précédent dans le temps.

PostgreSQL écrit les modifications de base de données dans des fichiers WAL qui sont uniquement ignorés après qu'AWS DMS a réussi à lire les modifications à partir de l'emplacement de réplication logique. L'utilisation d'emplacements de réplication logique permet de protéger les modifications consignées, pour qu'elles ne soient pas supprimées avant d'être consommées par le moteur de réplication.

Toutefois, selon le taux de changement et de consommation, les modifications conservées dans un emplacement de réplication logique peuvent entraîner une utilisation élevée du disque. Nous vous recommandons de définir des alarmes d'utilisation de l'espace dans l'instance PostgreSQL source lorsque vous utilisez des emplacements de réplication logique. Pour de plus amples informations sur l'utilisation de l'attribut de connexion supplémentaire `slotName`, veuillez consulter Paramètres du point de terminaison et attributs de connexion supplémentaires (ECA) lors de l'utilisation de PostgreSQL comme source DMS.

La procédure suivante décrit cette approche plus en détail.

Pour utiliser un point de départ CDC natif afin de configurer une charge CDC d'un point de terminaison source PostgreSQL

1. Identifiez l'emplacement de réplication logique utilisé par une tâche de réplication antérieure (tâche parent) que vous souhaitez utiliser comme point de départ. Ensuite, recherchez la vue `pg_replication_slots` de votre base de données source pour vous assurer que cet emplacement ne présente aucune connexion active. S'il en a, résolvez-les et fermez-les avant de poursuivre.

   Pour les étapes suivantes, il faut supposer que votre emplacement de réplication logique est `abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef`.
2. Créez un nouveau point de terminaison source qui inclut le paramètre d'attribut de connexion supplémentaire suivant.

   ```
   slotName=abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef;
   ```
3. Créez une nouvelle tâche de CDC uniquement à l'aide de la console, d'AWS CLI ou de l'API AWS DMS. Par exemple, à l'aide de l'interface de ligne de commande, vous pouvez exécuter la commande `create-replication-task` suivante.

```
aws dms create-replication-task --replication-task-identifier postgresql-slot-name-
test
--source-endpoint-arn arn:aws:dms:us-
west-2:012345678901:endpoint:ABCD1EFGHIJK2LMNOPQRST3UV4
--target-endpoint-arn arn:aws:dms:us-
west-2:012345678901:endpoint:ZYX9WVUTSRQONM8LKJIHGF7ED6
--replication-instance-arn arn:aws:dms:us-
west-2:012345678901:rep:AAAAAAAAAAA5BB4CCC3DDDD2EE
--migration-type cdc --table-mappings "file://mappings.json" --cdc-start-position
 "4AF/B00000D0"
--replication-task-settings "file://task-pg.json"
```

Dans la commande précédente, les options suivantes sont définies :

- L'option `source-endpoint-arn` est définie sur la nouvelle valeur que vous avez créée à l'étape 2.
- L'option `replication-instance-arn` est définie sur la même valeur que pour la tâche parent à l'étape 1.
- Les options `table-mappings` et `replication-task-settings` sont définies sur les mêmes valeurs que pour la tâche parent à l'étape 1.
- L'option `cdc-start-position` est définie sur une valeur de position de départ. Pour trouver cette position de départ, recherchez la vue `pg_replication_slots` de votre base de données source ou affichez les détails de la console pour la tâche parent à l'étape 1. Pour plus d'informations, consultez Définition d'un point de départ natif CDC.

Pour activer le mode de départ CDC personnalisé lors de la création d'une nouvelle tâche de CDC uniquement à l'aide de la console AWS DMS, procédez comme suit :

- Dans la section Paramètres de tâche, pour Mode de départ CDC pour les transactions sources, choisissez Activer le mode de départ CDC personnalisé.
- Pour Point de départ CDC personnalisé pour les transactions sources, choisissez Spécifier un numéro de séquence de journal. Spécifiez le numéro SCN ou choisissez Spécifier un point de contrôle de récupération et fournissez un point de contrôle de récupération.

Lorsque cette tâche CDC s'exécute, AWS DMS déclenche une erreur si l'emplacement de réplication logique spécifié n'existe pas. Il déclenche également une erreur si la tâche n'est pas créée avec un paramètre valide pour `cdc-start-position`.

Lorsque vous utilisez des points de départ CDC natifs avec le plug-in pglogical et que vous souhaitez utiliser un nouvel emplacement de réplication, effectuez les étapes de configuration suivantes avant de créer une tâche de CDC.

Pour utiliser un nouvel emplacement de réplication qui n'a pas été créé précédemment dans le cadre d'une autre tâche DMS

1. Créez un emplacement de réplication, comme indiqué ci-dessous :

```
SELECT * FROM pg_create_logical_replication_slot('replication_slot_name',
 'pglogical');
```

2. Une fois que la base de données a créé l'emplacement de réplication, obtenez les valeurs de restart_lsn et confirmed_flush_lsn pour l'emplacement et notez-les :

```
select * from pg_replication_slots where slot_name like 'replication_slot_name';
```

   Notez que la position de départ CDC native d'une tâche de CDC créée après l'emplacement de réplication ne peut pas être antérieure à la valeur de confirmed_flush_lsn.

   Pour en savoir plus sur les valeurs de restart_lsn et confirmed_flush_lsn, consultez pg_replication_slots.

3. Créez un nœud pglogical.

```
SELECT pglogical.create_node(node_name := 'node_name', dsn := 'your_dsn_name');
```

4. Créez deux ensembles de réplications à l'aide de la fonction `pglogical.create_replication_set`. Le premier ensemble de réplications assure le suivi des mises à jour et des suppressions des tables dotées de clés primaires. Le deuxième ensemble de réplications assure uniquement le suivi des insertions et porte le même nom que le premier ensemble de réplications, avec le préfixe « i ».

```
SELECT pglogical.create_replication_set('replication_slot_name', false, true, true,
 false);
SELECT pglogical.create_replication_set('ireplication_slot_name', true, false,
 false, true);
```

5. Ajoutez une table à l'ensemble de réplications.

```
SELECT pglogical.replication_set_add_table('replication_slot_name',
 'schemaname.tablename', true);
SELECT pglogical.replication_set_add_table('ireplication_slot_name',
 'schemaname.tablename', true);
```

6. Définissez l'attribut de connexion supplémentaire (ECA) suivant lorsque vous créez le point de terminaison source.

```
PluginName=PGLOGICAL;slotName=slot_name;
```

Vous pouvez désormais créer une tâche de CDC uniquement avec un point de départ natif PostgreSQL à l'aide du nouvel emplacement de réplication. Pour plus d'informations sur le plug-in pglogical, consultez la documentation de pglogical 3.7.

Migration de PostgreSQL vers PostgreSQL à l'aide d'AWS DMS.

Lorsque vous effectuez une migration à partir d'un moteur de base de données autre que PostgreSQL vers une base de données PostgreSQL, AWS DMS est, très souvent, le meilleur outil de migration à utiliser. En revanche, lorsque vous effectuez une migration à partir d'une base de données PostgreSQL vers une base de données PostgreSQL, les outils PostgreSQL peuvent être plus efficaces.

Utilisation des outils natifs PostgreSQL pour migrer des données

Nous vous recommandons d'utiliser les outils de migration de base de données PostgreSQL, par exemple pg_dump, dans les conditions suivantes :

- Vous avez une migration homogène, dans laquelle vous effectuez une migration d'une base de données PostgreSQL source vers une base de données PostgreSQL cible.
- Vous migrez une base de données entière.

- Les outils natifs vous permettent de migrer vos données avec une interruption minimale.

L'utilitaire pg_dump utilise la commande COPY pour créer un schéma et un vidage des données d'une base de données PostgreSQL. Le script de vidage généré par pg_dump charge les données dans une base de données portant le même nom et recrée les tables, les index et les clés étrangères. Pour restaurer les données dans une base de données portant un autre nom, utilisez la commande `pg_restore` et le paramètre `-d`.

Si vous migrez des données d'une base de données source PostgreSQL exécutée sur EC2 vers une cible Amazon RDS for PostgreSQL, vous pouvez utiliser le plug-in pglogical.

Pour plus d'informations sur l'importation d'une base de données PostgreSQL dans Amazon RDS for PostgreSQL ou Amazon Aurora PostgreSQL-Compatible Edition, consultez https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/PostgreSQL.Procedural.Importing.html.

Utilisation de DMS pour migrer des données de PostgreSQL vers PostgreSQL

AWS DMS peut migrer des données d'une base de données PostgreSQL source sur site, par exemple, vers une instance Amazon RDS for PostgreSQL ou Aurora PostgreSQL cible. La migration des types de données PostgreSQL principaux ou de base réussit souvent.

> **Note**
>
> Lorsque vous répliquez des tables partitionnées d'une source PostgreSQL vers une cible PostgreSQL, il n'est pas nécessaire de mentionner la table parent dans les critères de sélection de la tâche DMS. En mentionnant la table parent, cela déclenche la duplication des données dans les tables enfants de la cible, ce qui peut entraîner une violation de la PK. En sélectionnant uniquement les tables enfants dans les critères de sélection du mappage de table, la table parent est automatiquement remplie.

Les types de données pris en charge sur la base de données source mais pas sur la cible peuvent ne pas migrer avec succès. AWS DMS diffuse certains types de données sous forme de chaînes si le type de données est inconnu. La migration de certains types de données, tels que JSON et XML, peut réussir s'il s'agit de fichiers peu volumineux, mais peut échouer s'il s'agit de documents volumineux.

Lorsque vous effectuez une migration du type de données, prenez en considération ce qui suit :

- Dans certains cas, le type de données PostgreSQL NUMERIC(p,s) ne spécifie aucune précision ni aucune échelle. Pour DMS versions 3.4.2 et versions antérieures, DMS utilise une précision de 28 et une échelle de 6 par défaut, soit NUMERIC(28,6). Par exemple, la valeur 0,611111104488373 de la source est convertie en 0,611111 sur la cible PostgreSQL.
- Une table avec un type de données ARRAY doit posséder une clé primaire. Une table dont le type de données ARRAY ne contient pas de clé primaire est suspendue pendant le chargement complet.

Le tableau suivant montre les types de données PostgreSQL source et indique si leur migration peut réussir :

Type de données	Réussite de la migration	Migre partiellement	Ne migre pas	Commentaires
INTEGER	X			
SMALLINT	X			
BIGINT	X			
NUMERIC/DECIMAL(p,s)		X		Où 0<p<39 et 0<s
NUMERIC/DECIMAL		X		Où p>38 ou p=s=0
REAL	X			
DOUBLE	X			
SMALLSERIAL	X			
SERIAL	X			
BIGSERIAL	X			
MONEY	X			

Type de données	Réussite de la migration	Migre partiellement	Ne migre pas	Commentaires
CHAR		X		Sans précision spécifiée
CHAR(n)	X			
VARCHAR		X		Sans précision spécifiée
VARCHAR(n)	X			
TEXT	X			
BYTEA	X			
TIMESTAMP	X			Les valeurs infinies positives et négatives sont tronquées à « 9999-12-31 23:59:59 » et « 4713-01-01 00:00:00 BC » respectivement.
TIMESTAMP WITH TIME ZONE		X		
DATE	X			
TIME	X			

Type de données	Réussite de la migration	Migre partiellement	Ne migre pas	Commentaires
TIME WITH TIME ZONE		X		
INTERVAL		X		
BOOLEAN	X			
ENUM			X	
CIDR	X			
INET			X	
MACADDR			X	
TSVECTOR			X	
TSQUERY			X	
xml		X		
POINT	X			Type de données spatiales PostGIS
LINE			X	
LSEG			X	
BOX			X	
PATH			X	
POLYGON	X			Type de données spatiales PostGIS

Type de données	Réussite de la migration	Migre partiellement	Ne migre pas	Commentaires
CIRCLE			X	
JSON		X		
ARRAY	X			Nécessite une clé primaire
COMPOSITE			X	
RANGE			X	
LINESTRING	X			Type de données spatiales PostGIS
MULTIPOINT	X			Type de données spatiales PostGIS
MULTILINESTRING	X			Type de données spatiales PostGIS
MULTIPOLYGON	X			Type de données spatiales PostGIS
GEOMETRYCOLLECTION	X			Type de données spatiales PostGIS

Migration de types de données spatiales PostGIS

Les données spatiales identifient les informations géométriques d'un objet ou d'un emplacement dans l'espace. Les bases de données relationnelles objet PostgreSQL prennent en charge les types de données spatiales PostGIS.

Avant de migrer des objets de données spatiales PostgreSQL, assurez-vous que le plug-in PostGIS est activé au niveau global. Ce faisant, AWS DMS est sûr de créer les colonnes de données spatiales sources exactes pour l'instance de base de données cible PostgreSQL.

Pour les migrations homogènes PostgreSQL vers PostgreSQL, AWS DMS prend en charge la migration des types et sous-types d'objets de données géométriques et géographiques PostGIS (coordonnées géodésiques) géométriques et géographiques (coordonnées géodésiques) tels que les suivants :

- POINT
- LINESTRING
- POLYGON
- MULTIPOINT
- MULTILINESTRING
- MULTIPOLYGON
- GEOMETRYCOLLECTION

Suppression d'objets AWS DMS d'une base de données source PostgreSQL.

Pour capturer les événements DDL, AWS DMS crée divers artefacts dans la base de données PostgreSQL lorsqu'une tâche de migration démarre. Lorsque la tâche se termine, vous pouvez supprimer ces objets.

Pour supprimer les artefacts, émettez les instructions suivantes (dans leur ordre d'apparition), où `{AmazonRDSMigration}` est le schéma dans lequel les artefacts ont été créés. Procédez avec prudence lors de la suppression d'un schéma. Ne supprimez jamais un schéma opérationnel, surtout s'il est public.

```
drop event trigger awsdms_intercept_ddl;
```

Le déclencheur d'événements n'appartient pas à un schéma spécifique.

```
drop function {AmazonRDSMigration}.awsdms_intercept_ddl()
drop table {AmazonRDSMigration}.awsdms_ddl_audit
drop schema {AmazonRDSMigration}
```

Paramètres de configuration supplémentaires lors de l'utilisation d'une base de données PostgreSQL en tant que source DMS

Vous pouvez ajouter des paramètres de configuration supplémentaires lors de la migration des données d'une base de données PostgreSQL de deux façons :

- Vous pouvez ajouter des valeurs à l'attribut de connexion supplémentaire pour capturer les événements DDL et pour spécifier le schéma dans lequel les objets de base de données DDL opérationnelle sont créés. Pour plus d'informations, consultez Paramètres du point de terminaison et attributs de connexion supplémentaires (ECA) lors de l'utilisation de PostgreSQL comme source DMS.
- Vous pouvez remplacer les paramètres de chaîne de connexion. Choisissez cette option pour effectuer l'une des opérations suivantes :
  - Spécifiez les paramètres AWS DMS internes. Ces paramètres étant rarement nécessaires, ils ne sont pas présentés dans l'interface utilisateur.
  - Spécifiez des valeurs de transmission (passthru) pour le client de base de données spécifique. AWS DMS comprend des paramètres de transmission dans la chaîne de connexion transmise au client de base de données.
- En utilisant le paramètre au niveau de la table `REPLICA IDENTITY` dans PostgreSQL versions 9.4 et ultérieures, vous pouvez contrôler les informations écrites dans les journaux WAL. C'est notamment le cas pour les journaux WAL qui identifient les lignes mises à jour ou supprimées. `REPLICA IDENTITY FULL` enregistre les anciennes valeurs de toutes les colonnes dans la ligne. Utilisez `REPLICA IDENTITY FULL` avec précaution pour chaque table, car `FULL` génère une quantité supplémentaire de journaux WAL qui peuvent être inutiles. Pour plus d'informations, consultez ALTER TABLE-REPLICA IDENTITY.

Utilisation du paramètre de point de MapBooleanAsBoolean terminaison PostgreSQL

Vous pouvez utiliser les paramètres de point de terminaison PostgreSQL pour mapper un booléen en tant que booléen de votre source PostgreSQL à une cible Amazon Redshift. Par défaut, un type BOOLEAN est migré au format varchar(5). Vous pouvez indiquer à `MapBooleanAsBoolean`

d'autoriser PostgreSQL à migrer le type booléen en tant que booléen, comme illustré dans l'exemple suivant.

```
--postgre-sql-settings '{"MapBooleanAsBoolean": true}'
```

Notez que vous devez définir ce paramètre à la fois sur les points de terminaison sources et cibles pour qu'il prenne effet.

Étant donné que MySQL n'a pas de type BOOLEAN, utilisez une règle de transformation plutôt que ce paramètre lors de la migration de données BOOLEAN vers MySQL.

Paramètres du point de terminaison et attributs de connexion supplémentaires (ECA) lors de l'utilisation de PostgreSQL comme source DMS

Vous pouvez utiliser les paramètres des points de terminaison et les attributs de connexion supplémentaires (ECA) pour configurer votre base de données source PostgreSQL. Vous spécifiez les paramètres du point de terminaison lorsque vous créez le point de terminaison source à l'aide de la AWS DMS console ou à l'aide de la `create-endpoint` commande dans le AWS CLI, avec la syntaxe `--postgre-sql-settings '{"EndpointSetting": "value", ...}'` JSON.

Le tableau suivant indique les paramètres des points de terminaison et les ECA que vous pouvez utiliser avec PostgreSQL en tant que source.

Nom d'attribut	Description
`CaptureDDLs`	Pour capturer les événements DDL, AWS DMS crée divers objets dans la base de données PostgreSQL lorsque la tâche démarre. Vous pouvez ensuite supprimer ces objets comme décrit dans Suppression d'objets AWS DMS d'une base de données source PostgreSQL.. Si cette valeur est définie sur false, vous n'avez pas besoin de créer de tables ni de déclencheurs dans la base de données source. Heure à laquelle les événements DDL sont enregistrés. Valeur par défaut : `true` Valeurs valides : true/false

Nom d'attribut	Description
	Exemple : `--postgre-sql-settings '{"CaptureDDLs": true}'`
`ConsumeMonotonicEvents`	Permet de contrôler la façon dont les transactions monolithiques comportant des LSN en double sont répliquées. Lorsque ce paramètre est défini sur `false`, les événements comportant des LSN en double sont consommés et répliqués sur la cible. Lorsque ce paramètre est défini sur `true`, seul le premier événement est répliqué, tandis que les événements comportant des LSN en double ne sont ni consommés ni répliqués sur la cible. Valeur par défaut : `false` Valeurs valides : false/true Exemple : `--postgre-sql-settings '{"ConsumeMonotonicEvents": true}'`
`DdlArtifactsSchema`	Définit le schéma dans lequel les artefacts de base de données DDL opérationnels sont créés. Valeur par défaut : public Valeurs valides : string Exemple : `--postgre-sql-settings '{"DdlArtifactsSchema": "` *`xyzddlschema`* `"}'`
`ExecuteTimeout`	Définit le délai d'attente de déclaration client pour l'instance PostgreSQL, en secondes. La valeur par défaut est de 60 secondes. Exemple : `--postgre-sql-settings '{"ExecuteTimeout": 100}'`

Nom d'attribut	Description
`FailTasksOnLobTruncation`	Lorsque la valeur est définie sur `true`, elle provoque l'échec d'une tâche si la taille réelle d'une colonne LOB est supérieure à la valeur `LobMaxSize` spécifiée. Si une tâche est définie sur le mode LOB limité et que cette option a pour valeur true, la tâche échoue au lieu de tronquer les données LOB. Valeur par défaut : false Valeurs valides : booléen Exemple : `--postgre-sql-settings '{"FailTasksOnLobTruncation": true}'`
`fetchCacheSize`	Cet attribut de connexion supplémentaire (ECA) définit le nombre de lignes que le curseur va extraire pendant le chargement complet. En fonction des ressources disponibles dans l'instance de réplication, vous pouvez augmenter ou réduire la valeur. Valeur par défaut : `10000` Valeurs valides : nombre Exemple d'ECA : `fetchCacheSize=10000;`
`HeartbeatFrequency`	Définit la fréquence de pulsation du WAL (en minutes). Valeur par défaut : `5` Valeurs valides : nombre Exemple : `--postgre-sql-settings '{"HeartbeatFrequency": 1}'`

Nom d'attribut	Description
`HeartbeatSchema`	Définit le schéma dans lequel les artefacts de pulsation sont créés. Valeur par défaut : `public` Valeurs valides : string Exemple : `--postgre-sql-settings '{"HeartbeatSchema": "xyzheartbeatschema"}'`
`MapJsonbAsClob`	Par défaut, AWS DMS mappe JSONB à NCLOB. Vous pouvez indiquer à `MapJsonbAsClob` d'autoriser PostgreSQL à migrer le type JSONB en tant que CLOB. Exemple : `--postgre-sql-settings='{"MapJsonbAsClob": "true"}'`
`MapLongVarcharAs`	Par défaut, AWS DMS mappe VARCHAR à WSTRING. Vous pouvez indiquer à `MapLongVarcharAs` d'autoriser PostgreSQL à migrer le type VARCHAR(N) (où N est supérieur à 16 387) vers les types suivants : • WSTRING • CLOB • NCLOB Exemple : `--postgre-sql-settings='{"MapLongVarcharAs": "CLOB"}'`

Nom d'attribut	Description
`MapUnboundedNumericAsString`	Ce paramètre traite les colonnes contenant des types de données NUMERIC illimités comme STRING afin de réussir la migration sans perte de précision de la valeur numérique. Utilisez ce paramètre uniquement pour la réplication d'une source PostgreSQL vers une cible PostgreSQL ou pour des bases de données compatibles avec PostgreSQL. Valeur par défaut : `false` Valeurs valides : `false/true` Exemple : `--postgre-sql-settings '{"MapUnboundedNumericAsString": true}'` L'utilisation de ce paramètre peut entraîner une certaine dégradation des performances de réplication en raison de la transformation de la valeur numérique en chaîne, puis à nouveau en valeur numérique. Ce paramètre est compatible avec DMS versions 3.4.4 et ultérieures. **Note** Utilisez `MapUnboundedNumericAsString` uniquement dans les points de terminaison source et cible PostgreSQL ensemble. L'utilisation de `MapUnboundedNumericAsString` sur les points de terminaison PostgreSQL source limite la précision à 28 pendant la CDC. L'utilisation de `MapUnboundedNumericAsString` sur les points de terminaison cibles migre les données avec une précision de 28 et une échelle de 6.

Nom d'attribut	Description
	N'utilisez pas `MapUnboundedNumericAsString` avec des cibles autres que PostgreSQL.
`PluginName`	Spécifie le plugin à utiliser pour créer un emplacement de réplication. Valeurs valides : `pglogical` , `test_decoding` Exemple : `--postgre-sql-settings '{"PluginName": "test_decoding"}'`

Nom d'attribut	Description
SlotName	Définit le nom d'un emplacement de réplication logique précédemment créé pour une charge CDC de l'instance PostgreSQL source. Lorsqu'il est utilisé avec le paramètre de demande d'API AWS DMS CdcStartPosition , cet attribut permet également d'utiliser des points de départ CDC natifs. Avant de démarrer la tâche de chargement du CDC, DMS vérifie que l'emplacement de réplication logique spécifié existe. Il vérifie également que la tâche a été créée avec un paramètre valide de CdcStartPosition . Si l'emplacement spécifié n'existe pas ou si la tâche n'a pas de paramètre CdcStartPosition valide, DMS déclenche une erreur. Pour de plus amples informations sur la configuration du paramètre de demande CdcStartPosition , consultez Définition d'un point de départ natif CDC. Pour plus d'informations sur l'utilisation de CdcStartPosition , consultez la documentation relative aux opérations d'API CreateReplicationTask , StartReplicationTask et ModifyReplicationTask dans la Référence des API AWS Database Migration Service. Valeurs valides : string Exemple : --postgre-sql-settings '{"SlotName": "abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef"}'
unboundedVarcharMaxSize	Cet attribut de connexion supplémentaire (ECA) définit la taille maximale d'une colonne de données définie en tant que type VarChar sans spécificateur de longueur maximale. La valeur par défaut est de 8 000 octets. La valeur maximale est de 10485760 octets.

Limitations de l'utilisation d'une base de données PostgreSQL en tant que source DMS

Les limites suivantes s'appliquent lorsque vous utilisez PostgreSQL comme source pour AWS DMS :

- AWS DMS ne fonctionne pas avec Amazon RDS for PostgreSQL 10.4 ou Amazon Aurora PostgreSQL 10.4 en tant que source ou cible.
- Une table capturée doit posséder une clé primaire. Si une table est dépourvue de clé primaire, AWS DMS ignore les opérations d'enregistrements DELETE et UPDATE pour cette table. Consultez Activation de la capture des données de modification (CDC) à l'aide de la réplication logique pour obtenir une solution de contournement.

  Remarque : il n'est pas recommandé d'effectuer une migration sans clé primaire/index unique. Dans le cas contraire, des limitations supplémentaires s'appliquent, telles que la capacité d'application par lots « NO », le mode LOB complet, la validation des données et l'impossibilité de répliquer efficacement vers une cible Redshift.
- AWS DMS ignore une tentative de mettre à jour un segment de clé primaire. Dans ces cas, la cible identifie la mise à jour comme une mise à jour n'ayant mis à jour aucune ligne. Toutefois, comme les résultats de la mise à jour d'une clé primaire dans PostgreSQL sont imprévisibles, aucun enregistrement n'est écrit dans la table d'exceptions.
- AWS DMS ne prend pas en charge l'option d'exécution Start Process Changes from Timestamp.
- AWS DMS ne réplique pas les modifications résultant d'opérations de partition ou de sous-partition (`ADD`, `DROP` ou `TRUNCATE`).
- La réplication de plusieurs tables portant le même nom mais avec une casse différente (par exemple, table1, TABLE1 et Table1) peut entraîner un comportement imprévisible. En raison de ce problème, AWS DMS ne prend pas en charge ce type de la réplication.
- Dans la plupart des cas, AWS DMS prend en charge le traitement des modifications apportées aux instructions DDL CREATE, ALTER et DROP pour les tables. AWS DMS ne prend pas en charge ce traitement des modifications si les tables sont contenues dans un bloc de fonction ou de procédure interne ou dans d'autres constructions imbriquées.

  Par exemple, la modification suivante n'est pas capturée.

```
CREATE OR REPLACE FUNCTION attu.create_distributors1() RETURNS void
LANGUAGE plpgsql
AS $$
BEGIN
create table attu.distributors1(did serial PRIMARY KEY,name
```

```
varchar(40) NOT NULL);
END;
$$;
```

- Actuellement, les types de données `boolean` dans une source PostgreSQL sont migrés vers une cible SQL Server en tant que type de données `bit` avec des valeurs incohérentes. Pour contourner ce problème, pré-créez la table avec un type de données `VARCHAR(1)` pour la colonne (ou demandez à AWS DMS de créer la table). Ensuite, faites en sorte que le traitement en aval traite un « F » comme Faux et un « T » comme Vrai.
- AWS DMS ne prend pas en charge le traitement des opérations TRUNCATE.
- Le type de données LOB OID n'est pas migré vers la cible.
- AWS DMS prend en charge le type de données PostGIS pour les migrations homogènes uniquement.
- Si votre source est une base de données PostgreSQL sur site ou sur une instance Amazon EC2, assurez-vous que le plug-in de sortie test_decoding est installé sur le point de terminaison source. Vous trouverez ce plug-in dans le package contrib PostgreSQL. Pour plus d'informations sur le plug-in test_decoding, consultez la documentation PostgreSQL.
- AWS DMS ne prend pas en charge le traitement des modifications pour définir et annuler la définition des valeurs par défaut des colonnes (à l'aide de la clause ALTER COLUMN SET DEFAULT dans des instructions ALTER TABLE).
- AWS DMS ne prend pas en charge le traitement des modifications pour définir la possibilité de valeur NULL des colonnes (à l'aide de la clause ALTER COLUMN [SET|DROP] NOT NULL dans des instructions ALTER TABLE).
- Lorsque la réplication logique est activée, le nombre maximal de modifications conservées en mémoire par transaction est de 4 Mo. Ensuite, les modifications sont déversées sur le disque. En conséquence, `ReplicationSlotDiskUsage` augmente et `restart_lsn` n'avance pas tant que la transaction n'est pas terminée ou arrêtée et que la restauration n'est pas terminée. Comme il s'agit d'une transaction longue, sa restauration peut prendre beaucoup de temps. Évitez donc les transactions de longue durée ou les nombreuses sous-transactions lorsque la réplication logique est activée. Divisez plutôt la transaction en plusieurs transactions plus petites.

  Sur les versions 13 et ultérieures d'Aurora PostgreSQL, vous pouvez régler le paramètre pour contrôler `logical_decoding_work_mem` le moment où des déversements de DMS modifient les données sur le disque. Pour plus d'informations, consultez Déversez des fichiers dans Aurora PostgreSQL.

- Une table avec un type de données ARRAY doit posséder une clé primaire. Une table dont le type de données ARRAY ne contient pas de clé primaire est suspendue pendant le chargement complet.
- AWS DMS ne prend pas en charge la réplication des tables partitionnées. Lorsqu'une table partitionnée est détectée, voici ce qui se produit :
  - Le point de terminaison indique la liste des tables parent et enfant.
  - AWS DMS crée la table sur la cible en tant que table standard avec les mêmes propriétés que les tables sélectionnées.
  - Si la table parent dans la base de données source a la même valeur de clé primaire que ses tables enfants, une erreur « clé dupliquée » est générée.
- Afin de répliquer les tables partitionnés d'une source PostgreSQL vers une cible PostgreSQL, commencez par créer manuellement les tables parent et enfant sur la cible. Définissez ensuite une tâche distincte pour effectuer la réplication sur ces tables. Dans ce cas, définissez la configuration de la tâche sur Tronquer avant le chargement.
- Le type de données PostgreSQL `NUMERIC` n'est pas fixe. Lors du transfert de données de type `NUMERIC` sans précision ni échelle, DMS utilise `NUMERIC(28,6)` (une précision de 28 et une échelle de 6) par défaut. Par exemple, la valeur 0,611111104488373 de la source est convertie en 0,611111 sur la cible PostgreSQL.
- AWS DMS prend en charge Aurora PostgreSQL sans serveur V1 en tant que source pour les tâches de chargement complet uniquement. AWS DMS prend en charge Aurora PostgreSQL sans serveur V2 en tant que source pour les tâches de chargement complet, de chargement complet + CDC et de CDC uniquement.
- AWS DMS ne prend pas en charge la réplication d'une table contenant un index unique créé à l'aide d'une fonction coalesce.
- Lorsque vous utilisez le mode LOB, la table source et la table cible correspondante doivent avoir une clé primaire identique. Si l'une des tables ne contient pas de clé primaire, le résultat des opérations d'enregistrement DELETE et UPDATE sera imprévisible.
- Lorsque vous utilisez la fonctionnalité de chargement parallèle, la segmentation de table en fonction des partitions ou des sous-partitions n'est pas prise en charge. Pour plus d'informations sur le chargement parallèle, consultez Utilisation du chargement parallèle pour certaines tables, vues et collections.
- AWS DMS ne prend pas en charge les contraintes différées.
- AWS DMS version 3.4.7 prend en charge PostgreSQL 14.x en tant que source avec les limitations suivantes :

  - AWS DMS ne prend pas en charge le traitement des modifications lors des validations en deux phases.
  - AWS DMS ne prend pas en charge la réplication logique pour diffuser les transactions en cours de longue durée.
- AWS DMS ne prend pas en charge la CDC pour le proxy Amazon RDS pour PostgreSQL en tant que source.
- Lorsque vous utilisez des filtres de source qui ne contiennent pas de colonne de clé primaire, les opérations `DELETE` ne sont pas capturées.
- Si la base de données source est également la cible d'un autre système de réplication tiers, les modifications DDL risquent de ne pas être migrées pendant la CDC. En effet, cette situation peut empêcher le déclencheur d'événement `awsdms_intercept_ddl` de s'activer. Pour contourner ce problème, modifiez ce déclencheur dans la base de données source comme suit :

```
alter event trigger awsdms_intercept_ddl enable always;
```

- AWS DMS ne prend pas en charge la CDC pour le cluster de bases de données Amazon RDS Multi-AZ pour PostgreSQL en tant que source, car les clusters de bases de données RDS for PostgreSQL Multi-AZ ne prennent pas en charge la réplication logique.

Types de données sources pour PostgreSQL

Le tableau suivant présente les types de données sources PostgreSQL qui sont pris en charge lorsque vous utilisez AWS DMS et le mappage par défaut des types de données AWS DMS.

Pour en savoir plus sur la façon d'afficher le type de données qui est mappé dans la cible, consultez la section concernant le point de terminaison cible que vous utilisez.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Types de données PostgreSQL	Types de données DMS
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8

Types de données PostgreSQL	Types de données DMS
NUMERIC (p,s)	Si la précision se situe entre 0 et 38, utilisez NUMERIC. Si la précision est 39 ou supérieure, utilisez STRING.
DECIMAL(P,S)	Si la précision se situe entre 0 et 38, utilisez NUMERIC. Si la précision est 39 ou supérieure, utilisez STRING.
REAL	REAL4
DOUBLE	REAL8
SMALLSERIAL	INT2
SERIAL	INT4
BIGSERIAL	INT8
MONEY	NUMERIC(38,4) Le type de données MONEY est mappé sur FLOAT dans SQL Server.
CHAR	WSTRING (1)
CHAR(N)	WSTRING (n)
VARCHAR(N)	WSTRING (n)
TEXT	NCLOB
BYTEA	BLOB
TIMESTAMP	DATETIME

Types de données PostgreSQL	Types de données DMS
TIMESTAMP WITH TIME ZONE	DATETIME
DATE	DATE
TIME	TIME
TIME WITH TIME ZONE	TIME
INTERVAL	STRING (128) : 1 YEAR, 2 MONTHS, 3 DAYS, 4 HOURS, 5 MINUTES, 6 SECONDS
BOOLEAN	CHAR (5) Vrai ou faux ?
ENUM	STRING (64)
CIDR	STRING (50)
INET	STRING (50)
MACADDR	STRING (18)
BIT (n)	STRING (n)
BIT VARYING (n)	STRING (n)
UUID	CHAÎNE
TSVECTOR	CLOB
TSQUERY	CLOB
xml	CLOB
POINT	STRING (255) "(x,y)"
LINE	STRING (255) "(x,y,z)"
LSEG	STRING (255) "((x1,y1),(x2,y2))"
BOX	STRING (255) "((x1,y1),(x2,y2))"

Types de données PostgreSQL	Types de données DMS
PATH	CLOB "((x1,y1),(xn,yn))"
POLYGON	CLOB "((x1,y1),(xn,yn))"
CIRCLE	STRING (255) "(x,y),r"
JSON	NCLOB
JSONB	NCLOB
ARRAY	NCLOB
COMPOSITE	NCLOB
HSTORE	NCLOB
INT4RANGE	STRING (255)
INT8RANGE	STRING (255)
NUMRANGE	STRING (255)
STRRANGE	STRING (255)

Utilisation des types de données source LOB pour PostgreSQL

Les tailles de colonne PostgreSQL affectent la conversion des types de données LOB PostgreSQL en types de données AWS DMS. Pour utiliser cela, procédez comme suit pour les types de données AWS DMS :

- BLOB : définissez Limiter la taille LOB à sur la valeur Taille LOB maximale (Ko) lors de la création de la tâche.
- CLOB : la réplication gère chaque caractère en tant que caractère UTF8. Par conséquent, recherchez la longueur de la chaîne de caractères la plus longue dans la colonne, à savoir `max_num_chars_text` dans cet exemple. Utilisez cette longueur pour spécifier la valeur de Limiter la taille LOB à. Si les données comprennent des caractères à 4 octets, multipliez par 2 pour spécifier la valeur, en octets, de Limit LOB size to (Limiter la taille LOB à) Dans ce cas, la valeur de Limit LOB size to (Limiter la taille LOB à) est égale à `max_num_chars_text` multiplié par 2.

- NCLOB : la réplication gère chaque caractère en tant que caractère à deux octets. Par conséquent, recherchez la longueur de la chaîne de caractères la plus longue dans la colonne (`max_num_chars_text`) et multipliez-la par 2. Faites-le pour spécifier la valeur de Limiter la taille LOB à. Dans ce cas, la valeur de Limit LOB size to (Limiter la taille LOB à) est égale à `max_num_chars_text` multiplié par 2. Si les données comprennent des caractères à 4 octets, multipliez encore par 2. Dans ce cas, la valeur de Limit LOB size to (Limiter la taille LOB à) est égale à `max_num_chars_text` multiplié par 4.

Utilisation d'une base de données compatible MySQL comme source pour AWS DMS

Vous pouvez migrer les données d'une base de données compatible MySQL (MySQL, MariaDB ou Amazon Aurora MySQL) à l'aide d'AWS Database Migration Service.

Pour en savoir plus sur les versions de MySQL qui sont prises en charge par AWS DMS en tant que source, consultez Sources pour AWS DMS.

Vous pouvez utiliser SSL pour chiffrer les connexions entre votre point de terminaison compatible MySQL et l'instance de réplication. Pour plus d'informations sur l'utilisation de SSL avec un point de terminaison compatible MySQL, consultez Utilisation du protocole SSL avec AWS Database Migration Service.

Dans les sections suivantes, le terme « autogéré » s'applique à toute base de données installée sur site ou sur Amazon EC2. Le terme « géré par AWS » s'applique à toute base de données sur Amazon RDS, Amazon Aurora ou Amazon S3.

Pour plus de détails sur l'utilisation de bases de données compatibles MySQL et d'AWS DMS, consultez les sections suivantes.

Rubriques

- Migration de MySQL vers MySQL à l'aide d'AWS DMS.
- Utilisation de toute base de données compatible MySQL comme source pour AWS DMS
- Utilisation d'une base de données compatible MySQL autogérée comme source pour AWS DMS
- Utilisation d'une base de données compatible MySQL gérée par AWS en tant que source pour AWS DMS
- Limitations à l'utilisation d'une base de données MySQL comme source pour AWS DMS
- Prise en charge des transactions XA

- Paramètres de point de terminaison lors de l'utilisation de MySQL en tant que source pour AWS DMS
- Types de données sources pour MySQL

Migration de MySQL vers MySQL à l'aide d'AWS DMS.

Pour obtenir une migration hétérogène, dans laquelle vous effectuez une migration à partir d'un moteur de base de données autre que MySQL vers une base de données MySQL, AWS DMS est, très souvent, le meilleur outil de migration à utiliser. Mais pour une migration homogène, lorsque vous migrez d'une base de données MySQL vers une base de données MySQL, nous vous recommandons d'utiliser un projet de migrations de données homogènes. Les migrations de données homogènes utilisent des outils de base de données natifs pour fournir des performances et une précision de migration de données améliorées par rapport à AWS DMS.

Utilisation de toute base de données compatible MySQL comme source pour AWS DMS

Avant de commencer à utiliser une base de données MySQL comme source pour AWS DMS, veillez à ce que les conditions préalables suivantes soient remplies. Ces prérequis s'appliquent aux sources autogérées ou gérées par AWS.

Vous devez avoir un compte pour AWS DMS doté du rôle d'administrateur de réplication. Ce rôle nécessite les privilèges suivants :

- REPLICATION CLIENT : ce privilège est obligatoire pour les tâches de CDC uniquement. En d'autres termes, full-load-only les tâches ne nécessitent pas ce privilège.
- REPLICATION SLAVE : ce privilège est obligatoire pour les tâches de CDC uniquement. En d'autres termes, full-load-only les tâches ne nécessitent pas ce privilège.
- SUPER : ce privilège est nécessaire uniquement dans les versions de MySQL antérieures à 5.6.6.

L'utilisateur AWS DMS doit également disposer des privilèges SELECT pour les tables sources désignées pour la réplication.

Utilisation d'une base de données compatible MySQL autogérée comme source pour AWS DMS

Vous pouvez utiliser les bases de données compatibles MySQL autogérées suivantes comme sources pour AWS DMS :

- MySQL Community Edition
- MySQL Standard Edition
- MySQL Enterprise Edition
- MySQL Cluster Carrier Grade Edition
- MariaDB Community Edition
- MariaDB Enterprise Edition
- MariaDB Column Store

Pour utiliser la CDC, assurez-vous d'activer la journalisation binaire. Pour activer la journalisation binaire, les paramètres suivants doivent être configurés dans le fichier `my.ini` (Windows) ou `my.cnf` (UNIX) de MySQL.

Paramètre	Valeur
`server_id`	Définissez ce paramètre à une valeur 1 ou supérieure.
`log-bin`	Définissez le chemin d'accès au fichier journal binaire, par exemple `log-bin=E:\MySql_Logs\BinLog` . N'ajoutez pas d'extension de fichier.
`binlog_format`	Définissez ce paramètre à `ROW`. Nous recommandons d'utiliser ce paramètre lors de la réplication car, dans certains cas, lorsque `binlog_format` est défini sur `STATEMENT` , cela peut entraîner des incohérences lors de la réplication des données sur la cible. Le moteur de base de données écrit également des données incohérentes similaires sur la cible lorsque `binlog_format` est défini sur `MIXED`, car le moteur de base de données bascule automatiquement vers la journalisation basée sur `STATEMENT` , ce qui peut entraîner l'écriture de données incohérentes dans la base de données cible.
`expire_logs_days`	Définissez ce paramètre à une valeur 1 ou supérieure. Pour éviter l'utilisation excessive d'espace disque, nous recommandons de ne pas utiliser la valeur par défaut de 0.
`binlog_checksum`	Définissez ce paramètre sur `NONE` pour la version 3.4.7 ou antérieure de DMS.

Paramètre	Valeur
`binlog_row_image`	Définissez ce paramètre à `FULL`.
`log_slave_updates`	Définissez ce paramètre sur `TRUE` si vous utilisez un réplica en lecture MySQL ou MariaDB comme source.

Si votre source utilise le moteur de base de données NDB (en cluster), les paramètres suivants doivent être configurés pour permettre la capture des données modifiées sur les tables qui utilisent ce moteur de stockage. Ajoutez ces modifications dans le fichier `my.ini` (Windows) ou `my.cnf` (UNIX) de MySQL.

Paramètre	Valeur
`ndb_log_bin`	Définissez ce paramètre à `ON`. Cette valeur garantit que les modifications dans les tables en cluster sont journalisées dans le journal binaire.
`ndb_log_update_as_write`	Définissez ce paramètre à `OFF`. Cette valeur empêche d'écrire des instructions UPDATE en tant qu'instructions INSERT dans le journal binaire.
`ndb_log_updated_only`	Définissez ce paramètre à `OFF`. Cette valeur permet de s'assurer que le journal binaire contient l'ensemble de la ligne, pas seulement les colonnes modifiées.

Utilisation d'une base de données compatible MySQL gérée par AWS en tant que source pour AWS DMS

Vous pouvez utiliser les bases de données compatibles MySQL gérées par AWS suivantes en tant que sources pour AWS DMS :

- MySQL Community Edition
- MariaDB Community Edition
- Amazon Aurora MySQL-Compatible Edition

Lorsque vous utilisez une base de données compatible MySQL gérée par AWS en tant que source pour AWS DMS, veillez à ce que les prérequis suivants soient respectés pour la CDC :

- Pour activer les journaux binaires pour RDS for MySQL et pour RDS for MariaDB, activez les sauvegardes automatiques au niveau de l'instance. Pour activer les journaux binaires pour un cluster Aurora MySQL, modifiez la variable `binlog_format` dans le groupe de paramètres.

  Pour plus d'informations sur la configuration des sauvegardes automatiques, consultez Utilisation des sauvegardes dans le Guide de l'utilisateur Amazon RDS.

  Pour plus d'informations sur la configuration de la journalisation binaire pour une base de données Amazon RDS for MySQL, consultez Configuration de la journalisation binaire MySQL dans le Guide de l'utilisateur Amazon RDS.

  Pour plus d'informations sur la configuration de la journalisation binaire pour un cluster Aurora MySQL, consultez Comment puis-je activer la journalisation binaire pour mon cluster Amazon Aurora MySQL ?
- Si vous envisagez d'utiliser la CDC, activez la journalisation binaire. Pour plus d'informations sur la configuration de la journalisation binaire pour une base de données Amazon RDS for MySQL, consultez Configuration de la journalisation binaire MySQL dans le Guide de l'utilisateur Amazon RDS.
- Assurez-vous que les journaux binaires sont disponibles pour AWS DMS. Étant donné que les bases de données compatibles MySQL gérées par AWS purgent les journaux binaires dès que possible, vous devez augmenter la durée pendant laquelle les journaux demeurent disponibles. Par exemple, pour accroître la rétention des journaux à 24 heures, exécutez la commande suivante.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

- Définissez le paramètre `binlog_format` sur `"ROW"`.

> **Note**
>
> Sur MySQL ou MariaDB, `binlog_format` est un paramètre dynamique. Vous n'avez donc pas besoin de redémarrer pour que la nouvelle valeur prenne effet. Toutefois, la nouvelle valeur ne s'appliquera qu'aux nouvelles sessions. Si vous redéfinissez `binlog_format` sur `ROW` à des fins de réplication, la base de données peut toujours créer des journaux binaires ultérieurs en utilisant le format `MIXED`, si ces sessions ont débuté avant que

vous n'ayez modifié la valeur. Cela peut empêcher AWS DMS de capturer correctement toutes les modifications apportées à la base de données source. Lorsque vous modifiez le paramètre `binlog_format` sur une base de données MariaDB ou MySQL, veillez à redémarrer la base de données pour fermer toutes les sessions existantes, ou à redémarrer toute application effectuant des opérations DML (Data Manipulation Language). Le fait de forcer la base de données à redémarrer toutes les sessions après avoir redéfini le paramètre `binlog_format` sur `ROW` garantit que la base de données écrit toutes les modifications de base de données source suivantes dans le bon format. AWS DMS peut ainsi capturer correctement ces modifications.

- Définissez le paramètre `binlog_row_image` sur `"Full"`.
- Définissez le `binlog_checksum` paramètre sur `"NONE"` pour la version 3.4.7 ou antérieure de DMS. Pour plus d'informations sur la définition des paramètres dans Amazon RDS MySQL, consultez Utilisation des sauvegardes dans le Guide de l'utilisateur Amazon RDS.
- Si vous utilisez un réplica en lecture Amazon RDS MySQL ou Amazon RDS MariaDB en tant que source, activez les sauvegardes sur le réplica en lecture et assurez-vous de définir le paramètre `log_slave_updates` sur `TRUE`.

Limitations à l'utilisation d'une base de données MySQL comme source pour AWS DMS

Lorsque vous utilisez une base de données MySQL comme source, tenez compte des éléments suivants :

- La capture des données de modification (CDC) n'est pas prise en charge pour Amazon RDS MySQL 5.5 ou antérieur. Pour Amazon RDS MySQL, vous devez utiliser la version 5.6, 5.7 ou 8.0 pour activer la CDC. La CDC est prise en charge pour les sources MySQL 5.5 autogérées.
- Pour la CDC, `CREATE TABLE`, `ADD COLUMN`, `DROP COLUMN`, la modification du type de données de colonne et `renaming a column` sont pris en charge. Toutefois, `DROP TABLE`, `RENAME TABLE` et les mises à jour apportées à d'autres attributs, tels que la valeur par défaut de la colonne, la possibilité de valeur NULL de la colonne, le jeu de caractères, etc., ne sont pas pris en charge.
- Pour les tables partitionnées sur la source, lorsque vous définissez Target table preparation mode (Mode de préparation des tables cibles) sur Drop tables on target (Supprimer les tables sur la cible), AWS DMS crée une table simple sans partition sur la cible MySQL. Pour migrer des tables partitionnées vers une table partitionnée sur la cible, créez à l'avance les tables partitionnées sur la base de données MySQL cible.

- L'utilisation d'une instruction `ALTER TABLE table_name ADD COLUMN column_name` pour ajouter des colonnes au début (FIRST) ou au milieu d'une table (AFTER) n'est pas prise en charge. Les colonnes sont toujours ajoutées à la fin de la table.
- La capture des données modifiées (CDC) n'est pas prise en charge quand un nom de table contient des caractères minuscules et majuscules, et le moteur source est hébergé sur un système d'exploitation avec des noms de fichier sensibles à la casse. Un exemple est Microsoft Windows ou OS X utilisant HFS+.
- Vous pouvez utiliser Aurora MySQL-Compatible Edition sans serveur pour le chargement complet, mais vous ne pouvez pas l'utiliser pour la CDC. En effet, vous ne pouvez pas activer les prérequis pour MySQL. Pour plus d'informations , consultez Groupes de paramètres et Aurora sans serveur v1.
- L'attribut AUTO_INCREMENT sur une colonne n'est pas migré vers une colonne de base de données cible.
- La capture des modifications n'est pas prise en charge lorsque les journaux binaires ne sont pas stockés sur un stockage en bloc standard. Par exemple, la CDC ne fonctionne pas lorsque les journaux binaires sont stockés sur Amazon S3.
- AWS DMS crée des tables cibles avec le moteur de stockage InnoDB par défaut. Si vous avez besoin d'utiliser un moteur de stockage autre qu'InnoDB, vous devez créer manuellement la table et y migrer les données à l'aide du mode « Ne rien faire ».
- Vous ne pouvez pas utiliser les réplicas Aurora MySQL en tant que source pour AWS DMS, sauf si vous avez défini le mode de tâche de migration DMS sur Migrer les données existantes (chargement complet uniquement).
- Si la source compatible MySQL est arrêtée lors du chargement complet, la tâche AWS DMS ne s'arrête pas avec une erreur. La tâche se termine avec succès, mais la cible peut ne pas être synchronisée avec la source. Si cela se produit, redémarrez la tâche ou rechargez les tables affectées.
- Les index créés sur une partie d'une valeur de colonne ne sont pas migrés. Par exemple, l'index CREATE INDEX first_ten_chars ON customer (name(10)) n'est pas créé sur la cible.
- Dans certains cas, la tâche est configurée pour ne pas répliquer les LOB (la valeur SupportLobs « » est fausse dans les paramètres de la tâche ou l'option Ne pas inclure les colonnes LOB est sélectionnée dans la console des tâches). Dans ces cas, AWS DMS ne migre aucune colonne MEDIUMBLOB, LONGBLOB, MEDIUMTEXT ni LONGTEXT vers la cible.

  Les colonnes BLOB, TINYBLOB, TEXT et TINYTEXT ne sont pas affectées et sont migrées vers la cible.

- Les tables de données temporelles ou les tables gérées par version du système ne sont pas prises en charge dans les bases de données sources et cibles MariaDB.
- En cas de migration entre deux clusters Amazon RDS Aurora MySQL, le point de terminaison source RDS Aurora MySQL doit être une instance en lecture/écriture et non une instance de réplica.
- AWS DMS ne prend actuellement pas en charge la migration des vues pour MariaDB.
- AWS DMS ne prend pas en charge les modifications de DDL pour les tables partitionnées pour MySQL. Pour ignorer la suspension de table en cas de modification de DDL de partition pendant la CDC, définissez `skipTableSuspensionForPartitionDdl` sur `true`.
- AWS DMS prend uniquement en charge les transactions XA dans la versions 3.5.0 et ultérieures. Les versions précédentes ne prennent pas en charge les transactions XA. AWS DMS ne prend pas en charge les transactions XA dans MariaDB version 10.6. Pour plus d'informations, consultez the section called "Prise en charge des transactions XA", ci-après.
- AWS DMS n'utilise pas les GTID pour la réplication, même si les données source en contiennent.
- AWS DMS ne prend pas en charge la compression des transactions de journaux binaires.
- AWS DMS ne propage pas les événements ON DELETE CASCADE et ON UPDATE CASCADE pour les bases de données MySQL qui utilisent le moteur de stockage InnoDB. Pour ces événements, MySQL ne génère pas d'événements binlog pour refléter les opérations en cascade sur les tables enfants. Par conséquent, AWS DMS ne peut pas répliquer les modifications correspondantes dans les tables enfants. Pour plus d'informations, consultez Non-migration des index, des clés étrangères ou des mises à jour ou suppressions en cascade.
- AWS DMS ne capture pas les modifications apportées aux colonnes calculées (`VIRTUAL` et `GENERATED ALWAYS`). Pour contourner cette limitation, procédez comme suit :
  - Pré-créez la table cible dans la base de données cible et créez la tâche AWS DMS avec le paramètre de tâche de chargement complet `DO_NOTHING` ou `TRUNCATE_BEFORE_LOAD`.
  - Ajoutez une règle de transformation pour retirer la colonne calculée de la portée de tâche. Pour en savoir plus sur les règles de transformation, consultez Règles et actions de transformation.

Prise en charge des transactions XA

Une transaction Extended Architecture (XA) est une transaction qui peut être utilisée pour regrouper une série d'opérations provenant de plusieurs ressources transactionnelles en une seule transaction globale fiable. Une transaction XA utilise un protocole de validation en deux phases. En général, la capture des modifications avec des transactions XA ouvertes peut entraîner une perte de données.

Si la base de données n'utilise pas de transactions XA, vous pouvez ignorer cette autorisation et la configuration `IgnoreOpenXaTransactionsCheck` en utilisant la valeur par défaut `TRUE`. Pour commencer la réplication à partir d'une source qui contient des transactions XA, procédez comme suit :

- Assurez-vous que l'utilisateur de point de terminaison AWS DMS possède l'autorisation suivante :

```
grant XA_RECOVER_ADMIN on *.* to 'userName'@'%';
```

- Définissez le paramètre de point de terminaison `IgnoreOpenXaTransactionsCheck` sur `false`.

> **Note**
>
> AWS DMS ne prend pas en charge les transactions XA sur les bases de données sources MariaDB version 10.6.

Paramètres de point de terminaison lors de l'utilisation de MySQL en tant que source pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données source MySQL comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison source à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--my-sql-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec MySQL en tant que source sont indiqués dans le tableau suivant.

Name (Nom)	Description
`EventsPollInterval`	Spécifie la fréquence de vérification du journal binaire de nouvelles modifications/événements lorsque la base de données est inactif. Valeur par défaut : 5

Name (Nom)	Description
	Valeurs valides : 1 à 60 Exemple : `--my-sql-settings '{"EventsPollInterval": 5}'` Dans cet exemple, AWS DMS recherche des modificat ions dans les journaux binaires toutes les cinq secondes.
`ExecuteTimeout`	Pour AWS DMS versions 3.4.7 et ultérieures, définit le délai d'expiration de l'instruction client pour un point de terminaison source MySQL, en secondes. Valeur par défaut : 60 Exemple : `--my-sql-settings '{"ExecuteTimeout": 1500}'`
`ServerTimezone`	Spécifie le fuseau horaire de la base de données source MySQL. Exemple : `--my-sql-settings '{"ServerTimezone": "  US/Pacific "}'`
`AfterConnectScript`	Spécifie un script à exécuter immédiatement après la connexion de AWS DMS au point de terminaison. La tâche de migration continue à s'exécuter, que l'instruction SQL réussisse ou échoue. Valeurs valides : une ou plusieurs instructions SQL valides, séparées par un point-virgule. Exemple : `--my-sql-settings '{"AfterConnectScript": "ALTER SESSION SET CURRENT_SCHEMA=system"}'`

Name (Nom)	Description
`CleanSrcMetadataOnMismatch`	Nettoie et recrée les informations de métadonnées de table sur l'instance de réplication lorsqu'une non-correspondance survient. Par exemple, dans le cas où l'exécution d'une instruction alter DDL sur la table pourrait générer des informations différentes sur la table mise en cache dans l'instance de réplication. Booléen. Valeur par défaut : `false` Exemple : `--my-sql-settings '{"CleanSrcMetadataOnMismatch": false}'`
`skipTableSuspensionForPartitionDdl`	AWS DMS ne prend pas en charge les modifications de DDL pour les tables partitionnées pour MySQL. Pour AWS DMS les versions 3.4.6 et supérieures, le paramétrer pour `true` ignorer la suspension de table en cas de modification du DDL de partition pendant le CDC. AWS DMSignore le partitioned-table-related DDL et continue de traiter d'autres modifications du journal binaire. Valeur par défaut : `false` Exemple : `--my-sql-settings '{"skipTableSuspensionForPartitionDdl": true}'`
`IgnoreOpenXaTransactionsCheck`	Pour AWS DMS versions 3.5.0 et ultérieures, indique si les tâches doivent ignorer les transactions XA ouvertes lors du démarrage. Définissez ce paramètre sur `false` si votre source possède des transactions XA. Valeur par défaut : `true` Exemple : `--my-sql-settings '{"IgnoreOpenXaTransactionsCheck": false}'`

Types de données sources pour MySQL

Le tableau suivant présente les types de données sources de bases de données MySQL qui sont pris en charge lorsque vous utilisez AWS DMS et le mappage par défaut des types de données AWS DMS.

Pour en savoir plus sur la façon d'afficher le type de données qui est mappé dans la cible, consultez la section concernant le point de terminaison cible que vous utilisez.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Types de données MySQL	Types de données AWS DMS
INT	INT4
BIGINT	INT8
MEDIUMINT	INT4
TINYINT	INT1
SMALLINT	INT2
UNSIGNED TINYINT	UINT1
UNSIGNED SMALLINT	UINT2
UNSIGNED MEDIUMINT	UINT4
UNSIGNED INT	UINT4
UNSIGNED BIGINT	UINT8
DECIMAL(10)	NUMERIC (10,0)
BINAIRE	BYTES(1)
BIT	BOOLEAN
BIT(64)	BYTES(8)

Types de données MySQL	Types de données AWS DMS
BLOB	BYTES(65535)
LONGBLOB	BLOB
MEDIUMBLOB	BLOB
TINYBLOB	BYTES(255)
DATE	DATE
DATETIME	DATETIME DATETIME sans valeur entre parenthèses est répliqué sans millisecondes. DATETIME avec une valeur entre parenthèses comprise entre 1 et 5 (par exemple, `DATETIME(5)`) est répliqué en millisecondes. Lors de la réplication d'une colonne DATETIME, l'heure reste la même sur la cible. Elle n'est pas convertie au format UTC.
TIME	CHAÎNE
TIMESTAMP	DATETIME Lors de la réplication d'une colonne TIMESTAMP, l'heure est convertie au format UTC sur la cible.
YEAR	INT2
DOUBLE	REAL8

Types de données MySQL	Types de données AWS DMS
FLOAT	REAL(DOUBLE) Si les valeurs FLOAT ne sont pas comprises dans la plage suivante, utilisez une transform ation pour mapper FLOAT à STRING. Pour plus d'informations sur les transformations, consultez Règles et actions de transformation. La plage de valeurs FLOAT prise en charge est -1.79E+308 à -2.23E-308, 0 et 2.23E-308 à 1.79E+308
VARCHAR (45)	WSTRING (45)
VARCHAR (2000)	WSTRING (2000)
VARCHAR (4000)	WSTRING (4000)
VARBINARY (4000)	BYTES (4000)
VARBINARY (2000)	BYTES (2000)
CHAR	WSTRING
TEXT	WSTRING
LONGTEXT	NCLOB
MEDIUMTEXT	NCLOB
TINYTEXT	WSTRING(255)
GEOMETRY	BLOB
POINT	BLOB
LINESTRING	BLOB
POLYGON	BLOB

Types de données MySQL	Types de données AWS DMS
MULTIPOINT	BLOB
MULTILINESTRING	BLOB
MULTIPOLYGON	BLOB
GEOMETRYCOLLECTION	BLOB
ENUM	WSTRING (*longueur*) Ici, *longueur* correspond à la longueur de la valeur la plus longue dans ENUM.
SET	WSTRING (*longueur*) Ici, *longueur* correspond à la longueur totale de toutes les valeurs de SET, virgules incluses.
JSON	CLOB

> **Note**
>
> Dans certains cas, vous pouvez spécifier les types de données DATETIME et TIMESTAMP avec une valeur « zéro » (c'est-à-dire 0000-00-00). Dans ce cas, assurez-vous que la base de données cible de la tâche de réplication prend en charge les valeurs « zéro » pour les types de données DATETIME et TIMESTAMP. Dans le cas contraire, ces valeurs sont enregistrées en tant que valeurs NULL dans la cible.

Utilisation d'une base de données SAP ASE comme source pour AWS DMS

Vous pouvez migrer les données à partir d'une base de données SAP Adaptative Server Enterprise (ASE), anciennement Sybase, à l'aide d'AWS DMS. Avec une base de données SAP ASE comme source, vous pouvez migrer des données vers toute autre base de données cible AWS DMS prise en charge.

Pour en savoir plus sur les versions de SAP ASE qui sont prises en charge par AWS DMS en tant que source, consultez Sources pour AWS DMS.

Pour plus de détails sur l'utilisation des bases de données SAP ASE et d'AWS DMS, consultez les sections suivantes.

Rubriques

- Conditions préalables à l'utilisation d'une base de données SAP ASE comme source pour AWS DMS
- Limitations à l'utilisation de SAP ASE comme source pour AWS DMS
- Autorisations requises pour l'utilisation de SAP ASE comme source pour AWS DMS
- Suppression du point de troncature
- Paramètres de point de terminaison lors de l'utilisation de SAP ASE en tant que source pour AWS DMS
- Types de données sources pour SAP ASE

Conditions préalables à l'utilisation d'une base de données SAP ASE comme source pour AWS DMS

Pour qu'une base de données SAP ASE soit une source pour AWS DMS, effectuez les opérations suivantes :

- Activez la réplication SAP ASE pour les tables en utilisant la commande `sp_setreptable`. Pour plus d'informations, consultez les archives du centre de documentation Sybase (langue française non garantie).
- Désactivez `RepAgent` sur la base de données SAP ASE. Pour plus d'informations, consultez Arrêt et désactivation du thread RepAgent dans la base de données principale (langue française non garantie).
- Pour répliquer vers SAP ASE version 15.7 sur une instance Windows EC2 configurée pour les caractères non latins (par exemple, chinois), installez SAP ASE 15.7 SP121 sur l'ordinateur cible.

> **Note**
>
> Pour la réplication continue de la capture des données de modification (CDC), DMS exécute `dbcc logtransfer` et `dbcc log` pour lire les données du journal des transactions.

Limitations à l'utilisation de SAP ASE comme source pour AWS DMS

Les limites suivantes s'appliquent lors de l'utilisation d'une base de données SAP ASE comme source pour AWS DMS :

- Vous ne pouvez exécuter qu'une seule tâche AWS DMS avec la réplication continue ou la CDC pour chaque base de données SAP ASE. Vous pouvez exécuter plusieurs tâches de chargement complet uniquement en parallèle.
- Vous ne pouvez pas renommer une table. Par exemple, la commande suivante échoue.

```
sp_rename 'Sales.SalesRegion', 'SalesReg;
```

- Vous ne pouvez pas renommer une colonne. Par exemple, la commande suivante échoue.

```
sp_rename 'Sales.Sales.Region', 'RegID', 'COLUMN';
```

- Les valeurs zéro situées à la fin des chaînes de type de données binaires sont tronqués lors de la réplication vers la base de données cible. Par exemple, `0x00000000000000000000000001000001000000000` dans la table source devient `0x0000000000000000000000000100000100000001` dans la table cible.
- Si la base de données est configurée par défaut pour ne pas autoriser les valeurs NULL, AWS DMS crée la table cible avec des colonnes qui n'autorisent pas les valeurs NULL. Par conséquent, si une tâche de chargement complet ou de réplication des captures de données modifiées contient des valeurs vides, AWS DMS émet une erreur. Vous pouvez éviter ces erreurs en autorisant les valeurs NULL dans la base de données source à l'aide des commandes suivantes.

```
sp_dboption database_name, 'allow nulls by default', 'true'
go
use database_name
CHECKPOINT
go
```

- La commande d'index `reorg rebuild` n'est pas prise en charge.
- AWS DMS ne prend pas en charge les clusters ou MSA (Multi-Site Availability)/Secours semi-automatique en tant que source.
- Lorsque l'expression d'en-tête de transformation `AR_H_TIMESTAMP` est utilisée dans les règles de mappage, les millisecondes ne sont pas capturées pour une colonne ajoutée.

- L'exécution d'opérations de fusion pendant la CDC entraînera une erreur irrécupérable. Pour rétablir la synchronisation de la cible, effectuez un chargement complet.
- Les événements déclencheurs de restauration ne sont pas pris en charge pour les tables qui utilisent un schéma de verrouillage des lignes de données.
- AWS DMS ne peut pas reprendre une tâche de réplication après avoir supprimé une table dans la portée de la tâche depuis une base de données SAP source. Si la tâche de réplication DMS a été arrêtée et a effectué une opération DML (INSERT,UPDATE,DELETE) suivie de la suppression de la table, vous devez redémarrer la tâche de réplication.

Autorisations requises pour l'utilisation de SAP ASE comme source pour AWS DMS

Pour utiliser une base de données SAP ASE en tant que source dans une tâche AWS DMS, vous devez accorder des autorisations. Accordez au compte d'utilisateur spécifié dans les définitions de base de données AWS DMS les autorisations suivantes dans la base de données SAP ASE :

- sa_role
- replication_role
- sybase_ts_role
- Par défaut, lorsque vous devez être autorisé à exécuter la procédure stockée `sp_setreptable`, AWS DMS active l'option de réplication SAP ASE. Si vous souhaitez exécuter `sp_setreptable` sur une table directement à partir du point de terminaison de base de données sans passer par AWS DMS, vous pouvez utiliser l'attribut de connexion supplémentaire `enableReplication`. Pour de plus amples informations, veuillez consulter Paramètres de point de terminaison lors de l'utilisation de SAP ASE en tant que source pour AWS DMS.

Suppression du point de troncature

Lorsqu'une tâche démarre, AWS DMS crée une entrée `$replication_truncation_point` dans la vue du système `syslogshold`, ce qui indique qu'un processus de réplication est en cours. Pendant que AWS DMS fonctionne, il avance le point de troncature de réplication à intervalles réguliers, en fonction de la quantité de données qui a déjà été copiée dans la cible.

Une fois l'entrée `$replication_truncation_point` établie, maintenez la tâche AWS DMS en cours d'exécution pour empêcher le journal de base de données de devenir trop volumineux. Si vous souhaitez arrêter définitivement la tâche AWS DMS, supprimez le point de troncature de réplication en exécutant la commande suivante :

```
dbcc settrunc('ltm','ignore')
```

Une fois le point de troncature supprimé, vous ne pouvez pas reprendre la tâche AWS DMS. Le journal continue d'être tronqué automatiquement aux points de contrôle (si la troncature est configurée).

Paramètres de point de terminaison lors de l'utilisation de SAP ASE en tant que source pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données source SAP ASE comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison source à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--sybase-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec SAP ASE en tant que source sont indiqués dans le tableau suivant.

Name (Nom)	Description
`Charset`	Définissez cet attribut sur le nom SAP ASE correspon dant au jeu de caractères international. Valeur par défaut : `iso_1` Exemple : `--sybase-settings '{"Charset": "utf8"}'` Valeurs valides : • `acsii_8` • `big5hk` • `cp437` • `cp850` • `cp852` • `cp852` • `cp855`

Name (Nom)	Description
	• cp857 • cp858 • cp860 • cp864 • cp866 • cp869 • cp874 • cp932 • cp936 • cp950 • cp1250 • cp1251 • cp1252 • cp1253 • cp1254 • cp1255 • cp1256 • cp1257 • cp1258 • deckanji • euccns • eucgb • eucjis • eucksc • gb18030 • greek8 • iso_1 • iso88592 • iso88595

Name (Nom)	Description
	• `iso88596` • `iso88597` • `iso88598` • `iso88599` • `iso15` • `kz1048` • `koi8` • `roman8` • `iso88599` • `sjis` • `tis620` • `turkish8` • `utf8` Pour toute question supplémentaire sur les jeux de caractères pris en charge dans une base de données SAP ASE, consultez Adaptive Server Enterprise : Jeux de caractères pris en charge (langue française non garantie).
`EnableReplication`	Définissez cet attribut si vous souhaitez activer `sp_setreptable` sur les tables à partir de la base de données sans passer par AWS DMS. Valeur par défaut : `true` Valeurs valides : `true` ou `false` Exemple : `--sybase-settings '{"Enable Replication": false}'`

Name (Nom)	Description
`EncryptPassword`	Définissez cet attribut si vous avez activé `"net password encryption reqd"` dans la base de données source. Valeur par défaut : `0` Valeurs valides : `0`, 1 ou 2 Exemple : `--sybase-settings '{"EncryptPassword": 1}'` Pour plus d'informations sur ces valeurs de paramètres, consultez Adaptive Server Enterprise : Utilisation de la propriété de chaîne de connexion EncryptPassword (langue française non garantie).
`Provider`	Définissez cet attribut si vous souhaitez utiliser TLS (Transport Layer Security) 1.2 pour ASE 15.7 et versions ultérieures. Notez que AWS exige TLS version 1.2 ou ultérieure et recommande la version 1.3. Valeur par défaut : `Adaptive Server Enterprise` Valeurs valides : `Adaptive Server Enterprise 16.03.06` Exemple : `--sybase-settings '{"Provider": "Adaptive Server Enterprise 16.03.06"}'`

Types de données sources pour SAP ASE

Pour obtenir la liste des types de données sources SAP ASE pris en charge lorsque vous utilisez AWS DMS et le mappage par défaut des types de données AWS DMS, veuillez consulter le tableau suivant. AWS DMS ne prend pas en charge les tables sources SAP ASE ayant des colonnes de type UDT (type défini par l'utilisateur). Les colonnes répliquées avec ce type de données sont créées en tant que NULL.

Pour plus d'informations sur la façon d'afficher le type de données qui est mappé dans la cible, consultez la section Cibles pour la migration des données relative à votre point de terminaison cible.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Type de données SAP ASE	Types de données AWS DMS
BIGINT	INT8
UNSIGNED BIGINT	UINT8
INT	INT4
UNSIGNED INT	UINT4
SMALLINT	INT2
UNSIGNED SMALLINT	UINT2
TINYINT	UINT1
DECIMAL	NUMERIC
NUMERIC	NUMERIC
FLOAT	REAL8
DOUBLE	REAL8
REAL	REAL4
MONEY	NUMERIC
SMALLMONEY	NUMERIC
DATETIME	DATETIME
BIGDATETIME	DATETIME(6)
SMALLDATETIME	DATETIME

Type de données SAP ASE	Types de données AWS DMS
DATE	DATE
TIME	TIME
BIGTIME	TIME
CHAR	CHAÎNE
UNICHAR	WSTRING
NCHAR	WSTRING
VARCHAR	CHAÎNE
UNIVARCHAR	WSTRING
NVARCHAR	WSTRING
BINAIRE	BYTES
VARBINARY	BYTES
BIT	BOOLEAN
TEXT	CLOB
UNITEXT	NCLOB
IMAGE	BLOB

Utilisation de MongoDB comme source pour AWS DMS

Pour en savoir plus sur les versions de MongoDB qui sont prises en charge par AWS DMS en tant que source, consultez Sources pour AWS DMS.

Notez les éléments suivants concernant la prise en charge des versions de MongoDB :

- Les versions d'AWS DMS 3.4.5 et versions ultérieures prennent en charge MongoDB versions 4.2 et 4.4.

- Les versions d'AWS DMS 3.4.5 et versions ultérieures et les versions de MongoDB 4.2 et versions ultérieures prennent en charge les transactions distribuées. Pour plus d'informations sur les transactions distribuées MongoDB, consultez Transactions dans la documentation de MongoDB.
- Les versions d'AWS DMS 3.5.0 et versions ultérieures ne prennent pas en charge les versions de MongoDB antérieures à 3.6.
- Les versions d'AWS DMS 3.5.1 et versions ultérieures prennent en charge MongoDB version 5.0.
- Les versions 3.5.2 et ultérieures d'AWS DMS prennent en charge MongoDB version 6.0.

Si vous n'êtes pas encore familiarisé avec MongoDB, soyez conscient des concepts importants suivants relatifs aux bases de données MongoDB :

- Un enregistrement dans MongoDB est un document, c'est-à-dire une structure de données composée de paires champ-valeur. La valeur d'un champ peut contenir d'autres documents, tableaux et tableaux de documents. Un document correspond à peu près à une ligne dans une table de base de données relationnelle.
- Une collection dans MongoDB est un groupe de documents et équivaut à peu près à une table de base de données relationnelle.
- Une base de données dans MongoDB est un ensemble de collections et équivaut à peu près à un schéma dans une base de données relationnelle.
- Au niveau interne, un document MongoDB est stocké sous forme de fichier JSON binaire (BSON) dans un format compressé qui contient un type pour chaque champ du document. Chaque document présente un ID unique.

AWS DMS prend en charge deux modes de migration lorsque MongoDB est utilisé en tant que source : le mode document ou le mode table. Vous spécifiez le mode de migration à utiliser lorsque vous créez le point de terminaison MongoDB ou en définissant le paramètre Mode métadonnées à partir de la console AWS DMS. Vous pouvez éventuellement créer une deuxième colonne nommée `_id` qui fait office de clé primaire en cochant la case _id en tant que colonne séparée dans le panneau de configuration du point de terminaison.

Le mode de migration que vous choisissez affecte le format résultant des données cibles, comme indiqué ci-après.

Mode document

En mode document, le document MongoDB est migré tel quel, ce qui signifie que ses données sont regroupées en une seule colonne nommée `_doc` dans une table cible. Le mode document est défini par défaut lorsque MongoDB est utilisé comme point de terminaison source.

Prenons l'exemple des documents suivants dans une collection MongoDB appelée myCollection.

```
> db.myCollection.find()
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe0"), "a" : 1, "b" : 2, "c" : 3 }
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe1"), "a" : 4, "b" : 5, "c" : 6 }
```

Une fois les données migrées vers une table de base de données relationnelle à l'aide du mode document, les données sont structurées comme suit. Les champs de données du document MongoDB sont regroupés dans la colonne `_doc`.

oid_id	_doc
5a94815f40bd44d1b02bdfe0	`{ "a" : 1, "b" : 2, "c" : 3 }`
5a94815f40bd44d1b02bdfe1	`{ "a" : 4, "b" : 5, "c" : 6 }`

Vous pouvez éventuellement définir l'attribut de connexion supplémentaire `extractDocID` sur true pour créer une seconde colonne nommée `"_id"`, qui servira de clé primaire. Pour pouvoir utiliser CDC, définissez ce paramètre sur true.

En mode document, AWS DMS gère la création et le changement de nom des collections comme suit :

- Si vous ajoutez une nouvelle collection à la base de données source, AWS DMS crée une nouvelle table cible pour la collection et réplique tous les documents.
- Si vous renommez une collection existante dans la base de données source, AWS DMS ne renomme pas la table cible.

Si le point de terminaison cible est Amazon DocumentDB, exécutez la migration en mode document.

Mode table

En mode table, AWS DMS transforme chaque champ de niveau supérieur d'un document MongoDB en une colonne dans la table cible. Si un champ est imbriqué, AWS DMS aplatit les

valeurs imbriquées en une seule colonne. AWS DMS ajoute ensuite un champ clé et des types de données à l'ensemble de colonnes de la table cible.

Pour chaque document MongoDB, AWS DMS ajoute chaque clé et type à l'ensemble de colonnes de la table cible. Par exemple, en utilisant le mode table, AWS DMS migre l'exemple précédent dans la table suivante.

oid_id	a	b	c
5a94815f4 0bd44d1b02bdfe0	1	2	3
5a94815f4 0bd44d1b02bdfe1	4	5	6

Les valeurs imbriquées sont mises à plat dans une colonne contenant des noms de clé séparés par un point. La colonne est nommée en utilisant la forme concaténée des noms de champs mis à plat, séparés par des points. Par exemple, AWS DMS migre un document JSON avec un champ de valeurs imbriquées tel que `{"a" : {"b" : {"c": 1}}}` dans une colonne nommée `a.b.c.`

Pour créer les colonnes cibles, AWS DMS analyse un nombre spécifié de documents MongoDB et crée un ensemble regroupant tous les champs et leurs types. AWS DMS utilise ensuite cet ensemble pour créer les colonnes de la table cible. Si vous créez ou modifiez votre point de terminaison source MongoDB à l'aide de la console , vous pouvez spécifier le nombre de documents à analyser. La valeur par défaut est de 1000 documents. Si vous utilisez AWS CLI, vous pouvez utiliser l'attribut de connexion supplémentaire `docsToInvestigate`.

En mode table, AWS DMS gère les documents et les collections comme suit :

- Lorsque vous ajoutez un document à une collection existante, le document est répliqué. Si certains champs ne figurent pas dans la cible, ils ne sont pas répliqués.
- Lorsque vous mettez à jour un document, celui-ci est répliqué. Si certains champs ne figurent pas dans la cible, ils ne sont pas répliqués.
- La suppression d'un document est pleinement prise en charge.
- L'ajout d'une nouvelle collection n'entraîne pas la création d'une nouvelle table dans la cible lorsque cela est effectué durant une tâche CDC.

- Lors de la phase de capture des données de modification (CDC), AWS DMS ne permet pas de renommer une collection.

Rubriques

- Autorisations nécessaires lors de l'utilisation de MongoDB comme source pour AWS DMS
- Configuration d'un ensemble de réplicas MongoDB pour la CDC
- Exigences de sécurité lors de l'utilisation de MongoDB comme source pour AWS DMS
- Segmentation des collections MongoDB et migration en parallèle
- Migration de plusieurs bases de données lors de l'utilisation de MongoDB en tant que source pour AWS DMS
- Limites d'utilisation de MongoDB comme source pour AWS DMS
- Paramètres de configuration de point de terminaison lors de l'utilisation de MongoDB en tant que source pour AWS DMS
- Types de données sources pour MongoDB

Autorisations nécessaires lors de l'utilisation de MongoDB comme source pour AWS DMS

Pour une migration AWS DMS avec une source MongoDB, vous pouvez créer un compte utilisateur avec des privilèges racine ou un utilisateur avec des autorisations uniquement sur la base de données à migrer.

Le code suivant permet de créer un utilisateur qui sera le compte racine.

```
use admin
db.createUser(
  {
    user: "root",
    pwd: "password",
    roles: [ { role: "root", db: "admin" } ]
  }
)
```

Pour une source MongoDB 3.x, le code suivant permet de créer un utilisateur avec des autorisations minimales sur la base de données à migrer.

```
use database_to_migrate
db.createUser(
{
    user: "dms-user",
    pwd: "password",
    roles: [ { role: "read", db: "local" }, "read"]
})
```

Pour une source MongoDB 4.x, le code suivant permet de créer un utilisateur avec des privilèges minimum.

```
{ resource: { db: "", collection: "" }, actions: [ "find", "changeStream" ] }
```

Par exemple, créez le rôle suivant dans la base de données « admin ».

```
use admin
db.createRole(
{
role: "changestreamrole",
privileges: [
{ resource: { db: "", collection: "" }, actions: [ "find","changeStream" ] }
],
roles: []
}
)
```

Une fois le rôle créé, créez un utilisateur dans la base de données à migrer.

```
> use test
> db.createUser(
{
user: "dms-user12345",
pwd: "password",
roles: [ { role: "changestreamrole", db: "admin" }, "read"]
})
```

Configuration d'un ensemble de réplicas MongoDB pour la CDC

Pour utiliser la réplication continue ou la CDC avec MongoDB, AWS DMS a besoin d'accéder au journal des opérations MongoDB (oplog). Pour créer le journal oplog, vous devez déployer un

ensemble de réplicas, s'il n'en existe pas encore. Pour obtenir plus d'informations, consultez la documentation MongoDB.

Vous pouvez utiliser la capture des données modifiées (CDC) avec le nœud principal ou secondaire d'un ensemble de réplicas MongoDB comme point de terminaison source.

Pour convertir une instance autonome en un ensemble de réplicas

1. Utilisation d'une ligne de commande, se connecter à `mongo`.

   ```
   mongo localhost
   ```

2. Arrêtez le service `mongod`.

   ```
   service mongod stop
   ```

3. Redémarrez `mongod` à l'aide de la commande suivante :

   ```
   mongod --replSet "rs0" --auth -port port_number
   ```

4. Testez la connexion à l'ensemble de réplicas à l'aide des commandes suivantes :

   ```
   mongo -u root -p password --host rs0/localhost:port_number
     --authenticationDatabase "admin"
   ```

Si vous prévoyez d'effectuer une migration en mode document, sélectionnez l'option `_id as a separate column` lors de la création du point de terminaison MongoDB. La sélection de cette option permet de créer une seconde colonne nommée `_id`, qui agit en tant que clé primaire. Cette deuxième colonne est nécessaire pour qu'AWS DMS puisse prendre en charge les opérations de langage de manipulation de données (DML).

> **Note**
>
> AWS DMS utilise le journal des opérations (oplog) pour capturer les modifications pendant la réplication continue. Si MongoDB vide les enregistrements du journal des opérations avant qu'AWS DMS n'ait pu les lire, vos tâches échouent. Nous vous recommandons de dimensionner le journal des opérations pour conserver les modifications pendant au moins 24 heures.

Exigences de sécurité lors de l'utilisation de MongoDB comme source pour AWS DMS

AWS DMS prend en charge deux méthodes d'authentification pour MongoDB. Les deux méthodes d'authentification sont utilisées pour chiffrer le mot de passe. Elles sont donc utilisées uniquement lorsque le paramètre `authType` est défini sur PASSWORD.

Les méthodes d'authentification MongoDB sont les suivantes :

- MONGODB-CR : pour la rétrocompatibilité
- SCRAM-SHA-1 : méthode par défaut lors de l'utilisation de MongoDB version 3.x et 4.0

Si aucune méthode d'authentification n'est spécifiée, AWS DMS utilise la méthode par défaut pour la version de la source MongoDB.

Segmentation des collections MongoDB et migration en parallèle

Pour améliorer les performances d'une tâche de migration, les points de terminaison source MongoDB prennent en charge deux options de chargement complet en parallèle dans le mappage de table.

En d'autres termes, vous pouvez migrer une collection en parallèle en utilisant la segmentation automatique ou la segmentation par plage avec le mappage de table pour un chargement complet en parallèle dans les paramètres JSON. Avec la segmentation automatique, vous pouvez spécifier les critères permettant à AWS DMS de segmenter automatiquement votre source pour la migration dans chaque thread. Avec la segmentation par plage, vous pouvez indiquer à AWS DMS la plage spécifique de chaque segment que DMS doit migrer dans chaque thread. Pour plus d'informations sur ces paramètres, consultez Règles des paramètres de table et de collection et opérations.

Migration d'une base de données MongoDB en parallèle à l'aide de plages de segmentation automatique

Vous pouvez migrer vos documents en parallèle en spécifiant les critères permettant à AWS DMS de partitionner (segmenter) automatiquement vos données pour chaque thread. Vous spécifiez notamment le nombre de documents à migrer par thread. En utilisant cette approche, AWS DMS tente d'optimiser les limites de segment pour augmenter les performances par thread.

Vous pouvez spécifier les critères de segmentation à l'aide des options table-settings suivantes dans table-mapping.

Option table-settings	Description
`"type"`	(Obligatoire) Définie sur `"partitions-auto"` pour MongoDB en tant que source.
`"number-of-partitions"`	(Facultatif) Nombre total de partitions (segments) utilisées pour la migration. La valeur par défaut est 16.
`"collection-count-from-metadata"`	(Facultatif) Si cette option est définie sur `true`, AWS DMS utilise une estimation du nombre de collections pour déterminer le nombre de partitions. Si elle est définie sur `false`, AWS DMS utilise le nombre réel de collections. L'argument par défaut est `true`.
`"max-records-skip-per-page"`	(Facultatif) Nombre d'enregistrements à ignorer simultanément lors de la détermination des limites pour chaque partition. AWS DMS ignore un certain nombre d'enregistrements par page pour déterminer la limite minimale d'une partition . La valeur par défaut est 10 000. La définition d'une valeur relativement élevée peut entraîner des délais d'expiration du curseur et des échecs de tâche. La définition d'une valeur relativement faible entraîne un plus grand nombre d'opérations par page et ralentit le chargement complet.
`"batch-size"`	(Facultatif) Limite le nombre de documents renvoyés dans un lot. Chaque lot nécessite un aller retour jusqu'au serveur. Si la taille du lot est égale à zéro (0), le curseur utilise la taille de lot maximale définie par le serveur. La valeur par défaut est 0.

L'exemple suivant illustre un mappage de table pour la segmentation automatique.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "admin",
                "table-name": "departments"
            },
            "rule-action": "include",
            "filters": []
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "admin",
                "table-name": "departments"
            },
            "parallel-load": {
                "type": "partitions-auto",
                "number-of-partitions": 5,
                "collection-count-from-metadata": "true",
                "max-records-skip-per-page": 1000000,
                "batch-size": 50000
            }
        }
    ]
}
```

La segmentation automatique présente les limitations suivantes. Pour chaque segment, la migration extrait séparément le nombre de collections et la valeur minimale de `_id` pour la collection. Elle ignore ensuite un certain nombre d'enregistrements par page pour calculer la limite minimale pour ce segment.

Par conséquent, assurez-vous que la valeur minimale de `_id` reste constante pour chaque collection, jusqu'à ce que toutes les limites de segment de la collection soient calculées. Si vous modifiez la

valeur minimale de `_id` d'une collection lors du calcul de ses limites de segment, cela peut entraîner une perte de données ou des erreurs de lignes en double.

Migration d'une base de données MongoDB en parallèle à l'aide de la segmentation par plage

Vous pouvez migrer vos documents en parallèle en spécifiant les plages pour chaque segment d'un thread. En utilisant cette approche, vous indiquez à AWS DMS les documents à migrer dans chaque thread en fonction des plages de documents que vous avez choisies par thread.

L'image suivante illustre une collection MongoDB composée de sept éléments, qui utilise `_id` comme clé primaire.

Key	Value	Type
(1) ObjectId("5f805c74873173399a278d78")	{ 3 fields }	Object
_id	ObjectId("5f805c74873173399a278d78")	ObjectId
num	1	Int32
name	a	String
(2) ObjectId("5f805c97873173399a278d79")	{ 3 fields }	Object
_id	ObjectId("5f805c97873173399a278d79")	ObjectId
num	2	Int32
name	b	String
(3) ObjectId("5f805cb0873173399a278d7a")	{ 3 fields }	Object
_id	ObjectId("5f805cb0873173399a278d7a")	ObjectId
num	3	Int32
name	c	String
(4) ObjectId("5f805cbb873173399a278d7b")	{ 3 fields }	Object
_id	ObjectId("5f805cbb873173399a278d7b")	ObjectId
num	4	Int32
name	d	String
(5) ObjectId("5f805cc5873173399a278d7c")	{ 3 fields }	Object
_id	ObjectId("5f805cc5873173399a278d7c")	ObjectId
num	5	Int32
name	e	String
(6) ObjectId("5f805cd0873173399a278d7d")	{ 3 fields }	Object
_id	ObjectId("5f805cd0873173399a278d7d")	ObjectId
num	6	Int32
name	f	String
(7) ObjectId("5f805cdd873173399a278d7e")	{ 3 fields }	Object
_id	ObjectId("5f805cdd873173399a278d7e")	ObjectId
num	7	Int32
name	g	String

Pour diviser la collection en trois segments spécifiques qu'AWS DMS va migrer en parallèle, vous pouvez ajouter des règles de mappage de table à votre tâche de migration. Cette approche est illustrée dans l'exemple de code JSON suivant.

```
{ // Task table mappings:
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "testdatabase",
        "table-name": "testtable"
      },
      "rule-action": "include"
    }, // "selection" :"rule-type"
    {
      "rule-type": "table-settings",
      "rule-id": "2",
      "rule-name": "2",
      "object-locator": {
        "schema-name": "testdatabase",
        "table-name": "testtable"
      },
      "parallel-load": {
        "type": "ranges",
        "columns": [
           "_id",
           "num"
        ],
        "boundaries": [
          // First segment selects documents with _id less-than-or-equal-to
 5f805c97873173399a278d79
          // and num less-than-or-equal-to 2.
          [
             "5f805c97873173399a278d79",
             "2"
          ],
          // Second segment selects documents with _id > 5f805c97873173399a278d79 and
          // _id less-than-or-equal-to 5f805cc5873173399a278d7c and
          // num > 2 and num less-than-or-equal-to 5.
          [
             "5f805cc5873173399a278d7c",
```

```
            "5"
          ]
          // Third segment is implied and selects documents with _id >
 5f805cc5873173399a278d7c.
        ] // :"boundaries"
      } // :"parallel-load"
    } // "table-settings" :"rule-type"
  ] // :"rules"
} // :Task table mappings
```

Cette définition de mappage de table divise la collection source en trois segments et la migre en parallèle. Les limites de segmentation sont les suivantes.

```
Data with _id less-than-or-equal-to "5f805c97873173399a278d79" and num less-than-or-
equal-to 2 (2 records)
Data with _id > "5f805c97873173399a278d79" and num > 2 and _id  less-than-or-equal-to
 "5f805cc5873173399a278d7c" and num less-than-or-equal-to 5 (3 records)
Data with _id > "5f805cc5873173399a278d7c" and num > 5 (2 records)
```

Une fois la tâche de migration terminée, vous pouvez vérifier dans les journaux de tâches que les tables sont chargées en parallèle, comme indiqué dans l'exemple suivant. Vous pouvez également vérifier la clause `find` de MongoDB utilisée pour décharger chaque segment de la table source.

```
[TASK_MANAGER    ] I:  Start loading segment #1 of 3 of table
 'testdatabase'.'testtable' (Id = 1) by subtask 1. Start load timestamp
 0005B191D638FE86  (replicationtask_util.c:752)

[SOURCE_UNLOAD   ] I:  Range Segmentation filter for Segment #0 is initialized.
 (mongodb_unload.c:157)

[SOURCE_UNLOAD   ] I:  Range Segmentation filter for Segment #0 is: { "_id" :
 { "$lte" : { "$oid" : "5f805c97873173399a278d79" } }, "num" : { "$lte" :
 { "$numberInt" : "2" } } }  (mongodb_unload.c:328)

[SOURCE_UNLOAD   ] I: Unload finished for segment #1 of segmented table
 'testdatabase'.'testtable' (Id = 1). 2 rows sent.
```

```
[TASK_MANAGER    ] I: Start loading segment #1 of 3 of table
 'testdatabase'.'testtable' (Id = 1) by subtask 1. Start load timestamp
 0005B191D638FE86 (replicationtask_util.c:752)

[SOURCE_UNLOAD   ] I: Range Segmentation filter for Segment #0 is initialized.
 (mongodb_unload.c:157)

[SOURCE_UNLOAD   ] I: Range Segmentation filter for Segment #0 is: { "_id" : { "$lte" :
 { "$oid" : "5f805c97873173399a278d79" } }, "num" : { "$lte" : { "$numberInt" :
 "2" } } } (mongodb_unload.c:328)

[SOURCE_UNLOAD   ] I: Unload finished for segment #1 of segmented table
 'testdatabase'.'testtable' (Id = 1). 2 rows sent.

[TARGET_LOAD     ] I: Load finished for segment #1 of segmented table
 'testdatabase'.'testtable' (Id = 1). 1 rows received. 0 rows skipped. Volume
 transfered 480.

[TASK_MANAGER    ] I: Load finished for segment #1 of table
 'testdatabase'.'testtable' (Id = 1) by subtask 1. 2 records transferred.
```

Actuellement, AWS DMS prend en charge les types de données MongoDB suivants sous forme de colonne de clé de segment :

- Double
- Chaîne
- ObjectId
- Entier 32 bits
- Entier 64 bits

Migration de plusieurs bases de données lors de l'utilisation de MongoDB en tant que source pour AWS DMS

AWS DMS versions 3.4.5 et ultérieures prennent en charge la migration de plusieurs bases de données en une seule tâche pour toutes les versions de MongoDB prises en charge. Si vous souhaitez migrer plusieurs bases de données, procédez comme suit :

1. Lorsque vous créez le point de terminaison source MongoDB, procédez de l'une des manières suivantes :

- Sur la page Créer un point de terminaison de la console DMS, assurez-vous que le Nom de base de données est vide sous Configuration du point de terminaison.
- À l'aide de la commande AWS CLI `CreateEndpoint`, affectez une valeur de chaîne vide au paramètre `DatabaseName` dans `MongoDBSettings`.

2. Pour chaque base de données que vous souhaitez migrer à partir d'une source MongoDB, spécifiez le nom de la base de données sous forme de nom de schéma dans le mappage de table de la tâche. Pour ce faire, utilisez la saisie guidée dans la console ou effectuez votre saisie directement dans le code JSON. Pour plus d'informations sur la saisie guidée, consultez Spécification des règles de sélection de table et de transformation à partir de la console. Pour plus d'informations sur le code JSON, consultez Règles et actions de sélection.

Vous pouvez par exemple spécifier le code JSON suivant pour migrer trois bases de données MongoDB.

Example Migrer toutes les tables dans un schéma

Le code JSON suivant permet de migrer toutes les tables à partir des bases de données `Customers`, `Orders` et `Suppliers` du point de terminaison source vers votre point de terminaison cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Customers",
                "table-name": "%"
            },
            "rule-action": "include",
            "filters": []
        },
        {
            "rule-type": "selection",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "Orders",
                "table-name": "%"
```

```
            },
            "rule-action": "include",
            "filters": []
        },
        {
            "rule-type": "selection",
            "rule-id": "3",
            "rule-name": "3",
            "object-locator": {
                "schema-name": "Inventory",
                "table-name": "%"
            },
            "rule-action": "include",
            "filters": []
        }
    ]
}
```

Limites d'utilisation de MongoDB comme source pour AWS DMS

Les limites suivantes s'appliquent lorsque MongoDB est utilisé comme source pour AWS DMS :

- En mode table, les documents d'une collection doivent être cohérents dans le type de données qu'ils utilisent pour la valeur du même champ. Par exemple, si un document d'une collection inclut `'{ a:{ b:value ... }'`, tous les documents de la collection qui font référence à *`value`* dans le champ `a.b` doivent utiliser le même type de données pour *`value`*, quel que soit l'endroit où il apparaît dans la collection.
- Lorsque l'option `_id` est définie comme une colonne distincte, la chaîne d'ID ne peut pas contenir plus de 200 caractères.
- Les clés de type de tableau et d'ID d'objet sont converties en colonnes dotées des préfixes `array` et `oid` en mode table.

  En interne, ces colonnes sont référencées avec les noms préfixés. Si vous utilisez des règles de transformation dans AWS DMS qui font référence à ces colonnes, assurez-vous de spécifier la colonne préfixée. Par exemple, vous spécifiez `${oid__id}` et pas `${_id}`, ou `${array__addresses}` et pas `${_addresses}`.
- Les noms de collection et les noms de clé ne peuvent pas contenir le symbole du dollar ($).

- AWS DMS ne prend pas en charge les collections dont un même champ contient à la fois des majuscules et des minuscules en mode table avec une cible RDBMS. Par exemple, AWS DMS n'autorise pas que deux collections soient nommées `Field1` et `field1`.
- Le mode table et le mode document présentent les limitations décrites précédemment.
- La migration en parallèle à l'aide de la segmentation automatique présente les limitations décrites ci-dessus.
- Les filtres de source ne sont pas pris en charge pour MongoDB.
- AWS DMS ne prend pas en charge les documents dont le niveau d'imbrication est supérieur à 97.
- AWS DMS ne prend pas en charge les fonctionnalités suivantes de MongoDB version 5.0 :
  - Repartitionnement en direct
  - Chiffrement au niveau des champs côté client (CSFLE)
  - Migration de collections de séries temporelles

> **Note**
>
> Une collection de séries temporelles migrée pendant la phase de chargement complet sera convertie en collection normale dans Amazon DocumentDB, car DocumentDB ne prend pas en charge les collections de séries temporelles.

Paramètres de configuration de point de terminaison lors de l'utilisation de MongoDB en tant que source pour AWS DMS

Lorsque vous configurez le point de terminaison source MongoDB, vous pouvez spécifier plusieurs paramètres de configuration de point de terminaison à l'aide de la console AWS DMS.

Le tableau suivant décrit les paramètres de configuration disponibles lorsque vous utilisez les bases de données MongoDB en tant que source AWS DMS.

Paramètre (attribut)	Valeurs valides	Valeur par défaut et description

Mode d'authentification	`"none"`
	
	`"password"`
		La valeur `"password"` invite à entrer un nom d'utilisateur et un mot de passe. Lorsque `"none"` est spécifié, les paramètres de nom d'utilisateur et de mot de passe ne sont pas utilisés.

Paramètre (attribut)	Valeurs valides	Valeur par défaut et description
Source d'authentification	Un nom de base de données MongoDB valide.	Nom de la base de données MongoDB que vous souhaitez utiliser pour valider vos informations d'identification pour l'authentification. La valeur par défaut est `"admin"`.

Mécanisme d'authentification	`"default"`
	`"mongodb_cr"`
	`"scram_sha_1"`
		Mécanisme d'authentification. La valeur `"default"` est `"scram_sha_1"` . Ce paramètre n'est pas utilisé lorsque la valeur `authType` est définie sur `"no"`.

Mode métadonnées	Document et table	Choisit le mode document ou le mode table.
Nombre de documents à numériser (`docsToInvestigate`)	Un nombre entier positif supérieur à `0`.	Utilisez cette option en mode table uniquement pour définir la définition de table cible.
_id en tant que colonne séparée	Case cochée	Case à cocher facultative qui permet de créer une seconde colonne nommée `_id` qui sert de clé primaire.
`socketTimeoutMS`	NOMBRE Attribut de connexion supplémentaire (ECA) uniquement.	Ce paramètre est exprimé en millisecondes et configure le délai de connexion pour les clients MongoDB. Si la valeur est inférieure ou égale à zéro, la valeur par défaut du client MongoDB est utilisée.

Paramètre (attribut)	Valeurs valides	Valeur par défaut et description

`UseUpdateLookUp`	boolean
	
	`true`
	
	`false`
		Lorsque ce paramètre est défini sur true, pendant les événements de mise à jour de CDC, AWS DMS copie l'intégralité du document mis à jour et le colle dans la cible. Lorsqu'il est défini sur false, AWS DMS utilise la commande de mise à jour MongoDB pour mettre à jour uniquement les champs modifiés dans le document sur la cible.

`ReplicateShardCollections`	boolean
	
	`true`
	
	`false`
		Lorsque sa valeur est true, AWS DMS réplique les données vers des collections de partitions. AWS DMS utilise ce paramètre uniquement si le point de terminais on cible est un cluster Elastic DocumentDB.
		
		Lorsque ce paramètre est défini sur true, notez les éléments suivants :
		
		• Vous devez définir `TargetTablePrepMode` sur `nothing`.
		• AWS DMS définit automatiquement `useUpdateLookup` sur `false`.

Si vous choisissez Document comme Mode métadonnées, différentes options sont disponibles.

Si le point de terminaison cible est DocumentDB, assurez-vous d'exécuter la migration en Mode document. Modifiez également le point de terminaison source et sélectionnez l'option _id en tant que colonne séparée. Il s'agit d'un prérequis obligatoire si votre charge de travail MongoDB source implique des transactions.

Types de données sources pour MongoDB

La migration des données qui utilise MongoDB comme source pour AWS DMS prend en charge la plupart des types de données MongoDB. Dans le tableau suivant, vous pouvez trouver les types de données sources MongoDB qui sont pris en charge lorsque vous utilisez AWS DMS et le mappage par défaut des types de données AWS DMS. Pour de plus amples informations sur les types de données MongoDB, veuillez consulter Types BSON dans la documentation MongoDB.

Pour obtenir des informations sur la façon d'afficher le type de données qui est mappé dans la cible, consultez la section relative au point de terminaison cible que vous utilisez.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Types de données MongoDB	Types de données AWS DMS
Booléen	Booléen
Binaire	BLOB
Date	Date
Horodatage	Date
Int	INT4
Long	INT8
Double	REAL8
Chaîne (UTF-8)	CLOB
Tableau	CLOB
OID	Chaîne
REGEX	CLOB
CODE	CLOB

Utilisation d'Amazon DocumentDB (compatible avec MongoDB) en tant que source pour AWS DMS

Pour en savoir plus sur les versions d'Amazon DocumentDB (compatible avec MongoDB) prises en charge par AWS DMS en tant que source, consultez Sources pour AWS DMS.

En utilisant Amazon DocumentDB en tant que source, vous pouvez migrer des données d'un cluster Amazon DocumentDB vers un autre cluster Amazon DocumentDB. Vous pouvez également migrer

des données d'un cluster Amazon DocumentDB vers l'un des autres points de terminaison cibles pris en charge par AWS DMS.

Si vous utilisez Amazon DocumentDB pour la première fois, prenez connaissance des concepts importants suivants relatifs aux bases de données Amazon DocumentDB :

- Un enregistrement dans Amazon DocumentDB est un document, une structure de données composée de paires champ-valeur. La valeur d'un champ peut contenir d'autres documents, tableaux et tableaux de documents. Un document correspond à peu près à une ligne dans une table de base de données relationnelle.
- Une collection dans Amazon DocumentDB est un groupe de documents et équivaut à peu près à une table de base de données relationnelle.
- Une base de données dans Amazon DocumentDB est un ensemble de collections et équivaut à peu près à un schéma dans une base de données relationnelle.

AWS DMS prend en charge deux modes de migration lors de l'utilisation d'Amazon DocumentDB en tant que source : le mode document et le mode table. Vous spécifiez le mode de migration lorsque vous créez le point de terminaison source Amazon DocumentDB dans la console AWS DMS à l'aide de l'option Mode de métadonnées ou de l'attribut de connexion supplémentaire `nestingLevel`. Vous trouverez ci-dessous une explication sur la manière dont le choix du mode de migration affecte le format des données cibles.

Mode document

En mode document, le document JSON est migré tel quel. Cela signifie que les données du document sont regroupées dans l'un des deux éléments suivants. Lorsque vous utilisez une base de données relationnelle en tant que cible, les données correspondent à une colonne unique nommée `_doc` dans une table cible. Lorsque vous utilisez une base de données non relationnelle en tant que cible, les données correspondent à un document JSON unique. Le mode document est le mode par défaut. Nous recommandons de l'utiliser lors de la migration vers une cible Amazon DocumentDB.

Prenons l'exemple des documents suivants dans une collection Amazon DocumentDB appelée `myCollection`.

```
> db.myCollection.find()
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe0"), "a" : 1, "b" : 2, "c" : 3 }
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe1"), "a" : 4, "b" : 5, "c" : 6 }
```

Une fois les données migrées vers une table de base de données relationnelle à l'aide du mode document, les données sont structurées comme suit. Les champs de données du document sont regroupés dans la colonne `_doc`.

oid_id	_doc
5a94815f40bd44d1b02bdfe0	`{ "a" : 1, "b" : 2, "c" : 3 }`
5a94815f40bd44d1b02bdfe1	`{ "a" : 4, "b" : 5, "c" : 6 }`

Vous pouvez éventuellement définir l'attribut de connexion supplémentaire `extractDocID` sur `true` pour créer une seconde colonne nommée `"_id"`, qui servira de clé primaire. Si vous souhaitez utiliser la capture des données de modification (CDC), définissez ce paramètre sur `true`, sauf lorsque vous utilisez Amazon DocumentDB en tant que cible.

> **Note**
>
> Si vous ajoutez une nouvelle collection à la base de données source, AWS DMS crée une nouvelle table cible pour la collection et réplique tous les documents.

Mode table

En mode table, AWS DMS transforme chaque champ de niveau supérieur d'un document Amazon DocumentDB en colonne dans la table cible. Si un champ est imbriqué, AWS DMS aplatit les valeurs imbriquées en une seule colonne. AWS DMS ajoute ensuite un champ clé et des types de données à l'ensemble de colonnes de la table cible.

Pour chaque document Amazon DocumentDB, AWS DMS ajoute chaque clé et chaque type à l'ensemble de colonnes de la table cible. Par exemple, en utilisant le mode table, AWS DMS migre l'exemple précédent dans la table suivante.

oid_id	a	b	c
5a94815f4 0bd44d1b02bdfe0	1	2	3

5a94815f4 0bd44d1b02bdfe1	4	5	6

Les valeurs imbriquées sont mises à plat dans une colonne contenant des noms de clé séparés par un point. La colonne est nommée en utilisant la forme concaténée des noms de champs mis à plat, séparés par des points. Par exemple, AWS DMS migre un document JSON avec un champ de valeurs imbriquées tel que `{"a" : {"b" : {"c": 1}}}` dans une colonne nommée `a.b.c`.

Pour créer les colonnes cibles, AWS DMS analyse un nombre spécifié de documents Amazon DocumentDB et crée un ensemble regroupant tous les champs et leurs types. AWS DMS utilise ensuite cet ensemble pour créer les colonnes de la table cible. Si vous créez ou modifiez le point de terminaison source Amazon DocumentDB à l'aide de la console, vous pouvez spécifier le nombre de documents à analyser. La valeur par défaut est de 1 000 documents. Si vous utilisez AWS CLI, vous pouvez utiliser l'attribut de connexion supplémentaire `docsToInvestigate`.

En mode table, AWS DMS gère les documents et les collections comme suit :

- Lorsque vous ajoutez un document à une collection existante, le document est répliqué. Si certains champs ne figurent pas dans la cible, ils ne sont pas répliqués.
- Lorsque vous mettez à jour un document, celui-ci est répliqué. Si certains champs ne figurent pas dans la cible, ils ne sont pas répliqués.
- La suppression d'un document est pleinement prise en charge.
- L'ajout d'une nouvelle collection n'entraîne pas la création d'une nouvelle table dans la cible lorsque cela est effectué durant une tâche CDC.
- Lors de la phase de capture des données de modification (CDC), AWS DMS ne permet pas de renommer une collection.

Rubriques

- Définition des autorisations pour utiliser Amazon DocumentDB en tant que source
- Configuration de CDC pour un cluster Amazon DocumentDB
- Connexion à Amazon DocumentDB à l'aide de TLS
- Création d'un point de terminaison source Amazon DocumentDB
- Segmentation des collections Amazon DocumentDB et migration en parallèle

- Migration de plusieurs bases de données lors de l'utilisation d'Amazon DocumentDB en tant que source pour AWS DMS
- Limites d'utilisation d'Amazon DocumentDB en tant que source pour AWS DMS
- Utilisation des paramètres de point de terminaison avec Amazon DocumentDB en tant que source
- Types de données sources pour Amazon DocumentDB

Définition des autorisations pour utiliser Amazon DocumentDB en tant que source

Lorsque vous utilisez une source Amazon DocumentDB pour la migration AWS DMS, vous pouvez créer un compte d'utilisateur avec des privilèges racine. Vous pouvez également créer un utilisateur doté d'autorisations uniquement pour la base de données à migrer.

Le code suivant permet de créer un utilisateur en tant que compte racine.

```
use admin
db.createUser(
  {
    user: "root",
    pwd: "password",
    roles: [ { role: "root", db: "admin" } ]
  })
```

Pour Amazon DocumentDB 3.6, le code suivant permet de créer un utilisateur doté de privilèges minimum sur la base de données à migrer.

```
use database_to_migrate
db.createUser(
{
    user: "dms-user",
    pwd: "password",
    roles: [ { role: "read", db: "db_name" }, "read"]
})
```

Pour Amazon DocumentDB 4.0 et versions ultérieures, AWS DMS utilise un flux de modifications à l'échelle du déploiement. Ici, le code suivant permet de créer un utilisateur doté de privilèges minimum.

```
db.createUser(
{
    user: "dms-user",
    pwd: "password",
    roles: [ { role: "readAnyDatabase", db: "admin" }]
})
```

Configuration de CDC pour un cluster Amazon DocumentDB

Pour utiliser la réplication continue ou CDC avec Amazon DocumentDB, AWS DMS exige un accès aux flux de modifications du cluster Amazon DocumentDB. Pour obtenir une description de la séquence chronologique des événements de mise à jour dans les collections et les bases de données de votre cluster, consultez Utilisation des flux de modifications dans le Guide du développeur Amazon DocumentDB.

Authentifiez-vous auprès de votre cluster Amazon DocumentDB à l'aide du shell MongoDB. Ensuite, exécutez la commande suivante pour activer les flux de modifications.

```
db.adminCommand({modifyChangeStreams: 1,
    database: "DB_NAME",
    collection: "",
    enable: true});
```

Cette approche active le flux de modifications pour toutes les collections de la base de données. Une fois les flux de modifications activés, vous pouvez créer une tâche de migration qui migre les données existantes tout en répliquant les modifications continues. AWS DMS continue de capturer et d'appliquer les modifications, même après le chargement des données en bloc. Les bases de données sources et cibles sont finalement synchronisées, ce qui réduit au maximum le temps d'indisponibilité d'une migration.

> **Note**
>
> AWS DMS utilise le journal des opérations (oplog) pour capturer les modifications pendant la réplication continue. Si Amazon DocumentDB vide les enregistrements du journal des opérations avant qu'AWS DMS n'ait pu les lire, vos tâches échouent. Nous vous recommandons de dimensionner le journal des opérations pour conserver les modifications pendant au moins 24 heures.

Connexion à Amazon DocumentDB à l'aide de TLS

Par défaut, un cluster Amazon DocumentDB récemment créé n'accepte que les connexions sécurisées qui utilisent le protocole TLS (Transport Layer Security). Lorsque TLS est activé, chaque connexion à Amazon DocumentDB nécessite une clé publique.

Vous pouvez récupérer la clé publique pour Amazon DocumentDB en téléchargeant le fichier `rds-combined-ca-bundle.pem` à partir d'un compartiment Amazon S3 hébergé par AWS. Pour plus d'informations sur le téléchargement de ce fichier, consultez Chiffrement des connexions à l'aide de TLS dans le Guide du développeur Amazon DocumentDB.

Après avoir téléchargé le fichier `rds-combined-ca-bundle.pem`, vous pouvez importer la clé publique qu'il contient dans AWS DMS. Les étapes suivantes expliquent comment procéder.

Pour importer votre clé publique à l'aide de la console AWS DMS

1. Connectez-vous à la AWS Management Console et choisissez AWS DMS.
2. Dans le volet de navigation, choisissez Certificates.
3. Sélectionnez Importer un certificat. La page Importer un nouveau certificat de CA s'affiche.
4. Dans la section Configuration de certificat, effectuez l'une des opérations suivantes :
   - Pour Identifiant de certificat, entrez un nom unique pour le certificat, par exemple `docdb-cert`.
   - Choisissez Choisir un fichier, accédez à l'emplacement où vous avez enregistré le fichier `rds-combined-ca-bundle.pem`, puis sélectionnez-le.
5. Choisissez Ajouter un certificat CA.

L'exemple suivant dans AWS CLI utilise la commande AWS DMS `import-certificate` pour importer le fichier `rds-combined-ca-bundle.pem` de clé publique.

```
aws dms import-certificate \
    --certificate-identifier docdb-cert \
    --certificate-pem file://./rds-combined-ca-bundle.pem
```

Création d'un point de terminaison source Amazon DocumentDB

Vous pouvez créer un point de terminaison source Amazon DocumentDB à l'aide de la console ou d'AWS CLI. Procédez comme suit avec la console.

Pour configurer un point de terminaison source Amazon DocumentDB à l'aide de la console AWS DMS

1. Connectez-vous à la AWS Management Console et choisissez AWS DMS.
2. Choisissez Points de terminaison dans le volet de navigation, puis Créer un point de terminaison.
3. Pour Identifiant de point de terminaison, indiquez un nom qui vous permet de l'identifier facilement, tel que `docdb-source`.
4. Pour Moteur source, choisissez Amazon DocumentDB (compatible avec MongoDB).
5. Pour Nom du serveur, entrez le nom du serveur sur lequel réside le point de terminaison de la base de données Amazon DocumentDB. Par exemple, vous pouvez entrer le nom DNS public de votre instance Amazon EC2, tel que `democluster.cluster-cjf6q8nxfefi.us-east-2.docdb.amazonaws.com`.
6. Pour Port, entrez 27017.
7. Pour SSL mode (Mode SSL), choisissez verify-full. Si vous avez désactivé SSL sur votre cluster Amazon DocumentDB, vous pouvez ignorer cette étape.
8. Pour Certificat CA, choisissez le certificat Amazon DocumentDB, `rds-combined-ca-bundle.pem`. Pour obtenir des instructions sur l'ajout de ce certificat, consultez Connexion à Amazon DocumentDB à l'aide de TLS.
9. Pour Nom de base de données, entrez le nom de la base de données à migrer.

Procédez comme suit avec l'interface de ligne de commande.

Pour configurer un point de terminaison source Amazon DocumentDB à l'aide d'AWS CLI

- Exécutez la commande AWS DMS `create-endpoint` suivante pour configurer un point de terminaison source Amazon DocumentDB, en remplaçant les espaces réservés par vos propres valeurs.

```
aws dms create-endpoint \
           --endpoint-identifier a_memorable_name \
           --endpoint-type source \
           --engine-name docdb \
```

```
            --username value \
            --password value \
            --server-name servername_where_database_endpoint_resides \
            --port 27017 \
            --database-name name_of_endpoint_database
```

Segmentation des collections Amazon DocumentDB et migration en parallèle

Pour améliorer les performances d'une tâche de migration, les points de terminaison sources Amazon DocumentDB prennent en charge deux options de la fonctionnalité de chargement complet en parallèle dans le mappage de table. En d'autres termes, vous pouvez migrer une collection en parallèle en utilisant les options de segmentation automatique ou de segmentation par plage du mappage de table pour un chargement complet en parallèle dans les paramètres JSON. Les options de segmentation automatique vous permettent de spécifier les critères permettant à AWS DMS de segmenter automatiquement votre source pour la migration dans chaque thread. Les options de segmentation par plage vous permettent d'indiquer à AWS DMS la plage spécifique de chaque segment que DMS doit migrer dans chaque thread. Pour plus d'informations sur ces paramètres, consultez Règles des paramètres de table et de collection et opérations.

Migration d'une base de données Amazon DocumentDB en parallèle à l'aide de plages de segmentation automatique

Vous pouvez migrer vos documents en parallèle en spécifiant les critères permettant à AWS DMS de partitionner (segmenter) automatiquement vos données pour chaque thread, notamment le nombre de documents à migrer par thread. En utilisant cette approche, AWS DMS tente d'optimiser les limites de segment pour augmenter les performances par thread.

Vous pouvez spécifier les critères de segmentation à l'aide des options table-settings suivantes dans table-mapping :

Option table-settings	Description
`"type"`	(Obligatoire) Définissez cette option sur `"partitions-auto"` pour Amazon DocumentDB en tant que source.

Option table-settings	Description
`"number-of-partitions"`	(Facultatif) Nombre total de partitions (segments) utilisées pour la migration. La valeur par défaut est 16.
`"collection-count-from-metadata"`	(Facultatif) Si cette option est définie sur `true`, AWS DMS utilise une estimation du nombre de collections pour déterminer le nombre de partitions. Si elle est définie sur `false`, AWS DMS utilise le nombre réel de collections. L'argument par défaut est `true`.
`"max-records-skip-per-page"`	(Facultatif) Nombre d'enregistrements à ignorer simultanément lors de la détermination des limites pour chaque partition. AWS DMS ignore un certain nombre d'enregistrements par page pour déterminer la limite minimale d'une partition . La valeur par défaut est 10 000. La définition d'une valeur relativement élevée peut entraîner des délais d'expiration du curseur et des échecs de tâche. La définition d'une valeur relativement faible entraîne un plus grand nombre d'opérati ons par page et ralentit le chargement complet.
`"batch-size"`	(Facultatif) Limite le nombre de documents renvoyés dans un lot. Chaque lot nécessite un aller retour jusqu'au serveur. Si la taille du lot est égale à zéro (0), le curseur utilise la taille de lot maximale définie par le serveur. La valeur par défaut est 0.

L'exemple suivant illustre un mappage de table pour la segmentation automatique.

```
{
    "rules": [
        {
```

```
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "admin",
                "table-name": "departments"
            },
            "rule-action": "include",
            "filters": []
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "admin",
                "table-name": "departments"
            },
            "parallel-load": {
                "type": "partitions-auto",
                "number-of-partitions": 5,
                "collection-count-from-metadata": "true",
                "max-records-skip-per-page": 1000000,
                "batch-size": 50000
            }
        }
    ]
}
```

La segmentation automatique présente les limitations suivantes. Pour chaque segment, la migration extrait séparément le nombre de collections et la valeur minimale de `_id` pour la collection. Elle ignore ensuite un certain nombre d'enregistrements par page pour calculer la limite minimale pour ce segment. Par conséquent, assurez-vous que la valeur minimale de `_id` reste constante pour chaque collection, jusqu'à ce que toutes les limites de segment de la collection soient calculées. Si vous modifiez la valeur minimale de `_id` d'une collection lors du calcul de ses limites de segment, cela peut entraîner une perte de données ou des erreurs de lignes en double.

Migration d'une base de données Amazon DocumentDB en parallèle à l'aide de plages de segments spécifiques

L'exemple suivant illustre une collection Amazon DocumentDB composée de sept éléments, qui utilise `_id` comme clé primaire.

Key	Value	Type
(1) ObjectId("5f805c74873173399a278d78")	{ 3 fields }	Object
_id	ObjectId("5f805c74873173399a278d78")	ObjectId
num	1	Int32
name	a	String
(2) ObjectId("5f805c97873173399a278d79")	{ 3 fields }	Object
_id	ObjectId("5f805c97873173399a278d79")	ObjectId
num	2	Int32
name	b	String
(3) ObjectId("5f805cb0873173399a278d7a")	{ 3 fields }	Object
_id	ObjectId("5f805cb0873173399a278d7a")	ObjectId
num	3	Int32
name	c	String
(4) ObjectId("5f805cbb873173399a278d7b")	{ 3 fields }	Object
_id	ObjectId("5f805cbb873173399a278d7b")	ObjectId
num	4	Int32
name	d	String
(5) ObjectId("5f805cc5873173399a278d7c")	{ 3 fields }	Object
_id	ObjectId("5f805cc5873173399a278d7c")	ObjectId
num	5	Int32
name	e	String
(6) ObjectId("5f805cd0873173399a278d7d")	{ 3 fields }	Object
_id	ObjectId("5f805cd0873173399a278d7d")	ObjectId
num	6	Int32
name	f	String
(7) ObjectId("5f805cdd873173399a278d7e")	{ 3 fields }	Object
_id	ObjectId("5f805cdd873173399a278d7e")	ObjectId
num	7	Int32
name	g	String

Pour diviser la collection en trois segments et la migrer en parallèle, vous pouvez ajouter des règles de mappage de table à votre tâche de migration, comme indiqué dans l'exemple JSON suivant.

```
{ // Task table mappings:
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "testdatabase",
        "table-name": "testtable"
      },
      "rule-action": "include"
```

```
    }, // "selection" :"rule-type"
    {
      "rule-type": "table-settings",
      "rule-id": "2",
      "rule-name": "2",
      "object-locator": {
        "schema-name": "testdatabase",
        "table-name": "testtable"
      },
      "parallel-load": {
        "type": "ranges",
        "columns": [
           "_id",
           "num"
        ],
        "boundaries": [
          // First segment selects documents with _id less-than-or-equal-to
 5f805c97873173399a278d79
          // and num less-than-or-equal-to 2.
          [
             "5f805c97873173399a278d79",
             "2"
          ],
          // Second segment selects documents with _id > 5f805c97873173399a278d79 and
          // _id less-than-or-equal-to 5f805cc5873173399a278d7c and
          // num > 2 and num less-than-or-equal-to 5.
          [
             "5f805cc5873173399a278d7c",
             "5"
          ]
          // Third segment is implied and selects documents with _id >
 5f805cc5873173399a278d7c.
       ] // :"boundaries"
      } // :"parallel-load"
    } // "table-settings" :"rule-type"
  ] // :"rules"
} // :Task table mappings
```

Cette définition de mappage de table divise la collection source en trois segments et la migre en parallèle. Les limites de segmentation sont les suivantes.

```
Data with _id less-than-or-equal-to "5f805c97873173399a278d79" and num less-than-or-
equal-to 2 (2 records)
Data with _id less-than-or-equal-to "5f805cc5873173399a278d7c" and num less-than-or-
equal-to 5 and not in (_id less-than-or-equal-to  "5f805c97873173399a278d79" and num
 less-than-or-equal-to 2) (3 records)
Data not in (_id less-than-or-equal-to "5f805cc5873173399a278d7c" and num less-than-or-
equal-to 5) (2 records)
```

Une fois la tâche de migration terminée, vous pouvez vérifier dans les journaux de tâches que les tables sont chargées en parallèle, comme indiqué dans l'exemple suivant. Vous pouvez également vérifier la clause `find` d'Amazon DocumentDB utilisée pour décharger chaque segment de la table source.

```
[TASK_MANAGER    ] I:  Start loading segment #1 of 3 of table
 'testdatabase'.'testtable' (Id = 1) by subtask 1. Start load timestamp
 0005B191D638FE86  (replicationtask_util.c:752)

[SOURCE_UNLOAD   ] I:  Range Segmentation filter for Segment #0 is initialized.
 (mongodb_unload.c:157)

[SOURCE_UNLOAD   ] I:  Range Segmentation filter for Segment #0 is: { "_id" :
 { "$lte" : { "$oid" : "5f805c97873173399a278d79" } }, "num" : { "$lte" :
 { "$numberInt" : "2" } } }  (mongodb_unload.c:328)

[SOURCE_UNLOAD   ] I: Unload finished for segment #1 of segmented table
 'testdatabase'.'testtable' (Id = 1). 2 rows sent.

[TASK_MANAGER    ] I: Start loading segment #1 of 3 of table
 'testdatabase'.'testtable' (Id = 1) by subtask 1. Start load timestamp
 0005B191D638FE86 (replicationtask_util.c:752)

[SOURCE_UNLOAD   ] I: Range Segmentation filter for Segment #0 is initialized.
 (mongodb_unload.c:157)

[SOURCE_UNLOAD   ] I: Range Segmentation filter for Segment #0 is: { "_id" : { "$lte" :
 { "$oid" : "5f805c97873173399a278d79" } }, "num" : { "$lte" : { "$numberInt" :
 "2" } } } (mongodb_unload.c:328)

[SOURCE_UNLOAD   ] I: Unload finished for segment #1 of segmented table
 'testdatabase'.'testtable' (Id = 1). 2 rows sent.
```

```
[TARGET_LOAD     ] I: Load finished for segment #1 of segmented table
 'testdatabase'.'testtable' (Id = 1). 1 rows received. 0 rows skipped. Volume
 transfered 480.

[TASK_MANAGER    ] I: Load finished for segment #1 of table
 'testdatabase'.'testtable' (Id = 1) by subtask 1. 2 records transferred.
```

Actuellement, AWS DMS prend en charge les types de données Amazon DocumentDB suivants sous forme de colonne de clé de segment :

- Double
- Chaîne
- ObjectId
- Entier 32 bits
- Entier 64 bits

Migration de plusieurs bases de données lors de l'utilisation d'Amazon DocumentDB en tant que source pour AWS DMS

AWS DMS versions 3.4.5 et ultérieures prend en charge la migration de plusieurs bases de données en une seule tâche uniquement pour Amazon DocumentDB versions 4.0 et ultérieures. Si vous souhaitez migrer plusieurs bases de données, procédez comme suit :

1. Lorsque vous créez le point de terminaison source Amazon DocumentDB :
   - Dans AWS Management Console pour AWS DMS, laissez le champ Nom de base de données vide sous Configuration du point de terminaison sur la page Créer un point de terminaison.
   - Dans le AWS Command Line Interface (AWS CLI), attribuez une valeur de chaîne vide au DatabaseNameparamètre dans DocumentDBSettings que vous spécifiez pour l'CreateEndpointaction.
2. Pour chaque base de données que vous souhaitez migrer à partir de ce point de terminaison source Amazon DocumentDB, attribuez à chaque base de données le nom d'un schéma dans table-mapping pour la tâche en utilisant la saisie guidée dans la console ou directement dans le code JSON. Pour plus d'informations sur la saisie guidée, consultez la description dans Spécification des règles de sélection de table et de transformation à partir de la console. Pour plus d'informations sur le code JSON, consultez Règles et actions de sélection.

Vous pouvez par exemple spécifier le code JSON suivant pour migrer trois bases de données Amazon DocumentDB.

Example Migrer toutes les tables dans un schéma

Le code JSON suivant permet de migrer toutes les tables à partir des bases de données `Customers`, `Orders` et `Suppliers` du point de terminaison source vers votre point de terminaison cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Customers",
                "table-name": "%"
            },
            "object-locator": {
                "schema-name": "Orders",
                "table-name": "%"
            },
            "object-locator": {
                "schema-name": "Inventory",
                "table-name": "%"
            },
            "rule-action": "include"
        }
    ]
}
```

Limites d'utilisation d'Amazon DocumentDB en tant que source pour AWS DMS

Les limites suivantes s'appliquent lorsqu'Amazon DocumentDB est utilisé en tant que source pour AWS DMS :

- Lorsque l'option `_id` est définie comme une colonne distincte, la chaîne d'ID ne peut pas contenir plus de 200 caractères.
- Les clés de type de tableau et d'ID d'objet sont converties en colonnes dotées des préfixes `array` et `oid` en mode table.

En interne, ces colonnes sont référencées avec les noms préfixés. Si vous utilisez des règles de transformation dans AWS DMS qui font référence à ces colonnes, assurez-vous de spécifier la colonne préfixée. Par exemple, spécifiez `${oid__id}` et pas `${_id}`, ou `${array__addresses}` et pas `${_addresses}`.

- Les noms de collection et les noms de clé ne peuvent pas contenir le symbole du dollar ($).
- Le mode table et le mode document ont les limitations décrites précédemment.
- La migration en parallèle à l'aide de la segmentation automatique présente les limitations décrites ci-dessus.
- Une source Amazon DocumentDB (compatible avec MongoDB) ne prend pas en charge l'utilisation d'un horodatage spécifique comme position de départ pour la capture des données de modification (CDC). Une tâche de réplication continue commence à capturer les modifications quel que soit l'horodatage.
- Lorsque vous utilisez DocumentDB (compatible avec MongoDB) en tant que source, DMS peut gérer un maximum de 250 enregistrements par seconde.
- AWS DMS ne prend pas en charge les documents dont le niveau d'imbrication est supérieur à 97.
- Les filtres de source ne sont pas pris en charge pour DocumentDB.
- AWS DMSne prend pas en charge la réplication CDC (capture des données de modification) pour DocumentDB en tant que source en mode cluster élastique.

Utilisation des paramètres de point de terminaison avec Amazon DocumentDB en tant que source

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données source Amazon DocumentDB comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison source à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--doc-db-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec Amazon DocumentDB en tant que source sont indiqués dans le tableau suivant.

Nom d'attribut	Valeurs valides	Valeur par défaut et description
`NestingLevel`	`"none"`	`"none"` : spécifiez `"none"` pour utiliser le mode document. Spécifiez `"one"` pour activer le mode table.

Nom d'attribut	Valeurs valides	Valeur par défaut et description
	`"one"`	

`ExtractDocID`	boolean
	
	`true`
	
	`false`
		`false` : utilisez cet attribut lorsque `NestingLevel` est défini sur `"none"`.
		
		Si la base de données cible est Amazon DocumentDB, définissez `'{"ExtractDocID": true}'` .

`DocsToInvestigate`	Un nombre entier positif supérieur à `0`.	`1000` : utilisez cet attribut lorsque `NestingLevel` est défini sur `"one"`.

`ReplicateShardCollections`	boolean
	
	`true`
	
	`false`
		Lorsque sa valeur est true, AWS DMS réplique les données vers des collections de partitions. AWS DMS utilise ce paramètre uniquement si le point de terminaison cible est un cluster Elastic DocumentDB.
		
		Lorsque ce paramètre est défini sur true, notez les éléments suivants :
		
		• Vous devez définir `TargetTablePrepMode` sur `nothing`.
		• AWS DMS définit automatiquement `useUpdateLookup` sur `false`.

Types de données sources pour Amazon DocumentDB

Le tableau suivant répertorie présente les types de données sources Amazon DocumentDB pris en charge lors de l'utilisation de AWS DMS. Vous trouverez également le mappage par défaut à partir des types de données AWS DMS dans ce tableau. Pour plus d'informations sur les types de données, consultez Types BSON (langue française non garantie) dans la documentation de MongoDB.

Pour obtenir des informations sur la façon d'afficher le type de données qui est mappé dans la cible, consultez la section relative au point de terminaison cible que vous utilisez.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Types de données Amazon DocumentDB	Types de données AWS DMS
Booléen	Booléen
Binaire	BLOB
Date	Date
Horodatage	Date
Int	INT4
Long	INT8
Double	REAL8
Chaîne (UTF-8)	CLOB
Tableau	CLOB
OID	Chaîne

Utilisation d'Amazon S3 en tant que source pour AWS DMS

Vous pouvez migrer les données à partir d'un compartiment Amazon S3 à l'aide d'AWS DMS. Pour ce faire, fournissez un accès à un compartiment Amazon S3 contenant un ou plusieurs fichiers de données. Dans ce compartiment S3, incluez un fichier JSON qui décrit la mise en correspondance entre les données et les tables de base de données de ces données dans ces fichiers.

Les fichiers de données source doivent être présents dans le compartiment Amazon S3 avant que le chargement complet ne démarre. Vous spécifiez le nom de compartiment à l'aide du paramètre `bucketName`.

Les fichiers de données sources doivent être au format .csv (valeurs séparées par des virgules). Nommez-les à l'aide de la convention de dénomination suivante. Dans cette convention, *`schemaName`* est le schéma source et *`tableName`* est le nom d'une table dans ce schéma.

```
/schemaName/tableName/LOAD001.csv
/schemaName/tableName/LOAD002.csv
/schemaName/tableName/LOAD003.csv
```

```
...
```

Par exemple, supposons que vos fichiers de données se trouvent dans `mybucket` au chemin Amazon S3 suivant.

```
s3://mybucket/hr/employee
```

Au moment du chargement, AWS DMS suppose que le nom du schéma source est `hr` et que le nom de la table source est `employee`.

Outre `bucketName` (qui est requis), vous pouvez éventuellement fournir un paramètre `bucketFolder` pour spécifier où AWS DMS doit rechercher les fichiers de données dans le compartiment Amazon S3. Pour continuer avec l'exemple précédent, si vous définissez `bucketFolder` sur `sourcedata`, AWS DMS lit les fichiers de données au chemin suivant.

```
s3://mybucket/sourcedata/hr/employee
```

Vous pouvez spécifier le délimiteur de colonne, le délimiteur de ligne, l'indicateur de valeur null et d'autres paramètres à l'aide d'attributs de connexion supplémentaires. Pour de plus amples informations, veuillez consulter Paramètres de point de terminaison pour Amazon S3 en tant que source pour AWS DMS.

Vous pouvez spécifier le propriétaire du compartiment et empêcher toute tricherie en utilisant le paramètre de point de terminaison Amazon S3 `ExpectedBucketOwner`, comme indiqué ci-dessous. Ensuite, lorsque vous soumettez une demande pour tester une connexion ou effectuer une migration, S3 compare l'ID de compte du propriétaire du compartiment au paramètre spécifié.

```
--s3-settings='{"ExpectedBucketOwner": "AWS_Account_ID"}'
```

Rubriques

- Définition de tables externes pour Amazon S3 en tant que source pour AWS DMS
- Utilisation de la CDC avec Amazon S3 en tant que source pour AWS DMS
- Conditions préalables lors de l'utilisation d'Amazon S3 en tant que source pour AWS DMS
- Limitations relatives à l'utilisation d'Amazon S3 en tant que source pour AWS DMS
- Paramètres de point de terminaison pour Amazon S3 en tant que source pour AWS DMS
- Types de données source pour Amazon S3

Définition de tables externes pour Amazon S3 en tant que source pour AWS DMS

En plus des fichiers de données, vous devez fournir une définition de table externe. Une définition de table externe est un document JSON qui décrit comment AWS DMS doit interpréter les données provenant d'Amazon S3. La taille maximale de ce document est de 2 Mo. Si vous créez un point de terminaison source à l'aide de la console de gestion AWS DMS, vous pouvez entrer le fichier JSON directement dans la zone de mappage de table. Si vous utilisez l'AWS Command Line Interface (AWS CLI) ou l'API AWS DMS pour effectuer des migrations, vous pouvez créer un fichier JSON pour spécifier la définition de table externe.

Supposons que vous disposez d'un fichier de données qui inclut les éléments suivants.

```
101,Smith,Bob,2014-06-04,New York
102,Smith,Bob,2015-10-08,Los Angeles
103,Smith,Bob,2017-03-13,Dallas
104,Smith,Bob,2017-03-13,Dallas
```

Voici un exemple de définition de table externe pour ces données.

```
{
    "TableCount": "1",
    "Tables": [
        {
            "TableName": "employee",
            "TablePath": "hr/employee/",
            "TableOwner": "hr",
            "TableColumns": [
                {
                    "ColumnName": "Id",
                    "ColumnType": "INT8",
                    "ColumnNullable": "false",
                    "ColumnIsPk": "true"
                },
                {
                    "ColumnName": "LastName",
                    "ColumnType": "STRING",
                    "ColumnLength": "20"
                },
                {
                    "ColumnName": "FirstName",
                    "ColumnType": "STRING",
                    "ColumnLength": "30"
```

```
                },
                {
                    "ColumnName": "HireDate",
                    "ColumnType": "DATETIME"
                },
                {
                    "ColumnName": "OfficeLocation",
                    "ColumnType": "STRING",
                    "ColumnLength": "20"
                }
            ],
            "TableColumnsTotal": "5"
        }
    ]
}
```

Les éléments figurant dans ce document JSON sont les suivants :

`TableCount` : nombre de tables source. Cet exemple comporte une seule table.

`Tables` : tableau comprenant un mappage JSON par table source. Cet exemple comporte une seule carte. Chaque carte comprend les éléments suivants :

- `TableName` : nom de la table source.
- `TablePath` : chemin dans votre compartiment Amazon S3 où AWS DMS peut trouver le fichier de chargement complet des données. Si une valeur `bucketFolder` est spécifiée, elle est ajoutée en préfixe au chemin.
- `TableOwner` : nom du schéma pour cette table.
- `TableColumns` : tableau d'un ou plusieurs mappages, chacun décrivant une colonne de la table source :
  - `ColumnName` : nom d'une colonne de la table source.
  - `ColumnType` : type de données de la colonne. Pour connaître les types de données valides, consultez Types de données source pour Amazon S3.
  - `ColumnLength` : nombre d'octets dans cette colonne. La longueur maximale d'une colonne est limitée à 2 147 483 647 octets (2 047 mégaoctets), car une source S3 ne prend pas en charge le mode FULL LOB. `ColumnLength` est valide pour les types de données suivants :
    - BYTE
    - CHAÎNE

- `ColumnNullable` : valeur booléenne égale à `true` si cette colonne peut contenir des valeurs NULL (par défaut, `false`).
- `ColumnIsPk` : valeur booléenne égale à `true` si cette colonne fait partie de la clé primaire (par défaut, `false`).
- `ColumnDateFormat` : format de date d'entrée pour une colonne de types DATE, TIME et DATETIME ; utilisé pour analyser une chaîne de données dans un objet de date. Les valeurs possibles incluent :

```
- YYYY-MM-dd HH:mm:ss
- YYYY-MM-dd HH:mm:ss.F
- YYYY/MM/dd HH:mm:ss
- YYYY/MM/dd HH:mm:ss.F
- MM/dd/YYYY HH:mm:ss
- MM/dd/YYYY HH:mm:ss.F
- YYYYMMdd HH:mm:ss
- YYYYMMdd HH:mm:ss.F
```

- `TableColumnsTotal` : nombre total de colonnes. Ce nombre doit correspondre au nombre d'éléments dans le tableau `TableColumns`.

Sauf spécification contraire, AWS DMS suppose que `ColumnLength` a pour valeur zéro.

> **Note**
>
> Dans les versions prises en charge d'AWS DMS, les données source S3 peuvent également contenir une colonne d'opération facultative comme première colonne avant la valeur de colonne `TableName`. Cette colonne d'opération identifie l'opération (`INSERT`) utilisée pour migrer les données vers un point de terminaison cible S3 lors d'un chargement complet.
> Si elle est présente, la valeur de cette colonne est le caractère initial du mot-clé de l'opération `INSERT` (`I`). Si cette valeur est spécifiée, cette colonne indique généralement que la source S3 a été créée par DMS en tant que cible S3 au cours d'une migration précédente.
> Dans les versions de DMS antérieures à 3.4.2, cette colonne n'était pas présente dans les données source S3 créées à partir d'un chargement complet DMS précédent. L'ajout de cette colonne aux données cible S3 permet au format de toutes les lignes écrites dans la cible S3 d'être cohérent si celles-ci sont écrites lors d'un chargement complet ou d'un chargement CDC. Pour plus d'informations sur les options pour le formatage des données S3

cible, consultez Indication des opérations de base de données source dans des données S3 migrées.

Pour une colonne de type NUMERIC, spécifiez la précision et l'échelle. La précision est le nombre total de chiffres dans un nombre et l'échelle est le nombre de chiffres à droite de la virgule. Vous utilisez les éléments `ColumnPrecision` et `ColumnScale` pour cela, comme indiqué ci-après.

```
...
    {
        "ColumnName": "HourlyRate",
        "ColumnType": "NUMERIC",
        "ColumnPrecision": "5"
        "ColumnScale": "2"
    }
...
```

Pour une colonne de type DATETIME dont les données contiennent des fractions de seconde, spécifiez l'échelle. L'échelle est le nombre de chiffres pour les fractions de seconde. Elle est comprise entre 0 et 9. Vous utilisez l'élément `ColumnScale` pour cela, comme indiqué ci-après.

```
...
{
      "ColumnName": "HireDate",
      "ColumnType": "DATETIME",
      "ColumnScale": "3"
}
...
```

Si vous ne le spécifiez pas, AWS DMS suppose que `ColumnScale` est égal à zéro et tronque les fractions de seconde.

Utilisation de la CDC avec Amazon S3 en tant que source pour AWS DMS

Une fois qu'AWS DMS a effectué un chargement complet des données, il peut éventuellement répliquer les modifications des données au point de terminaison cible. Pour ce faire, vous chargez les fichiers de capture des données de modification (fichiers CDC) dans votre compartiment Amazon S3. AWS DMS lit ces fichiers CDC lorsque vous les chargez, puis applique les modifications au point de terminaison cible.

Les fichiers CDC sont nommés comme suit :

```
CDC00001.csv
CDC00002.csv
CDC00003.csv
...
```

> **Note**
>
> Pour répliquer des fichiers CDC dans le dossier des données de modification, chargez-les dans un ordre lexical (séquentiel). Par exemple, chargez le fichier CDC00002.csv avant le fichier CDC00003.csv. Dans le cas contraire, si vous chargez le fichier CDC00002.csv après le fichier CDC00003.csv, le premier est ignoré et n'est pas répliqué. Mais le fichier CDC00004.csv est répliqué avec succès s'il est chargé après CDC00003.csv.

Pour indiquer où AWS DMS peut trouver ces fichiers, spécifiez le paramètre `cdcPath`. Pour continuer avec l'exemple précédent, si vous définissez `cdcPath` sur *`changedata`*, AWS DMS lit les fichiers CDC correspondant au chemin suivant.

```
s3://mybucket/changedata
```

Si vous définissez `cdcPath` sur *`changedata`* et `bucketFolder` sur *`myFolder`*, AWS DMS lit les fichiers CDC au chemin suivant.

```
s3://mybucket/myFolder/changedata
```

Les enregistrements figurant dans un fichier CDC sont formatés comme suit :

- Opération : opération de modification à effectuer : `INSERT` ou `I`, `UPDATE` ou `U`, `DELETE` ou `D`. Ces mot-clé et ces valeurs de caractères ne sont pas sensibles à la casse.

  > **Note**
  >
  > Dans les versions d'AWS DMS prises en charge, AWS DMS peut identifier l'opération à effectuer pour chaque enregistrement de chargement de deux manières. AWS DMS peut le faire à partir de la valeur de mot-clé de l'enregistrement (par exemple, `INSERT`) ou à partir du caractère initial du mot-clé (par exemple, `I`). Dans les versions précédentes, AWS

DMS reconnaissait l'opération de chargement uniquement à partir de la valeur de mot-clé complète.
Pour les versions précédentes d'AWS DMS, la valeur de mot-clé complète était écrite pour journaliser les données de CDC. De plus, les versions précédentes écrivaient la valeur d'opération sur toute cible S3 à l'aide du mot-clé initial uniquement.
La reconnaissance des deux formats permet à AWS DMS de gérer l'opération, quelle que soit la façon dont la colonne d'opération est écrite pour créer les données source S3. Cette approche prend en charge l'utilisation de données cible S3 comme source pour une migration ultérieure. Avec cette approche, vous n'avez pas besoin de modifier le format de la valeur initiale d'un mot-clé qui apparaît dans la colonne d'opération de la source S3 ultérieure.

- Nom de la table : nom de la table source.
- Nom du schéma : nom du schéma source.
- Données : une ou plusieurs colonnes qui représentent les données à modifier.

Voici un exemple de fichier CDC pour une table nommée `employee`.

```
INSERT,employee,hr,101,Smith,Bob,2014-06-04,New York
UPDATE,employee,hr,101,Smith,Bob,2015-10-08,Los Angeles
UPDATE,employee,hr,101,Smith,Bob,2017-03-13,Dallas
DELETE,employee,hr,101,Smith,Bob,2017-03-13,Dallas
```

Conditions préalables lors de l'utilisation d'Amazon S3 en tant que source pour AWS DMS

Pour utiliser Amazon S3 en tant que source pour AWS DMS, votre compartiment S3 source doit se trouver dans la même région AWS que l'instance de réplication DMS qui migre vos données. De plus, le compte AWS que vous utilisez pour la migration doit avoir un accès en lecture au compartiment source.

Le rôle AWS Identity and Access Management (IAM) assigné au compte utilisateur utilisé pour créer une tâche de migration doit bénéficier de l'ensemble d'autorisations suivant.

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
```

```
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::mybucket*/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::mybucket*"
            ]
        }
    ]
}
```

Le rôle AWS Identity and Access Management (IAM) attribué au compte d'utilisateur utilisé pour créer la tâche de migration doit bénéficier de l'ensemble d'autorisations suivant si la gestion des versions est activée sur le compartiment Amazon S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "S3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::mybucket*/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::mybucket*"
            ]
        }
    ]
}
```

Limitations relatives à l'utilisation d'Amazon S3 en tant que source pour AWS DMS

Les limitations suivantes s'appliquent lorsque vous utilisez Amazon S3 en tant que source :

- N'activez pas la gestion des versions pour S3. Si vous avez besoin de la gestion des versions S3, utilisez des politiques de cycle de vie pour supprimer les anciennes versions de manière active. Dans le cas contraire, la connexion de test du point de terminaison risque d'échouer en raison du délai d'attente d'appel `list-object` S3. Pour créer une politique de cycle de vie pour un compartiment S3, consultez Gestion du cycle de vie de votre stockage. Pour supprimer une version d'un objet S3, consultez Suppression des versions d'objet d'un compartiment activé pour la gestion des versions.
- Un compartiment S3 compatible VPC (VPC de passerelle) est pris en charge dans les versions 3.4.7 et ultérieures.

Paramètres de point de terminaison pour Amazon S3 en tant que source pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données source Amazon S3 comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison source à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--s3-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec Amazon S3 en tant que source sont indiqués dans le tableau suivant.

Option	Description
`BucketFolder`	(Facultatif) Un nom de dossier dans le compartiment S3. Si cet attribut est spécifié, les fichiers de données source et les fichiers CDC sont lus à partir du chemin `s3://myBucket/bucketFolder /schemaNam e /tableName /` et `s3://myBucket/bucketFolder /`, respectiv

Option	Description
	ement. Si cet attribut n'est pas spécifié, le chemin utilisé est *schemaNam e* /*tableName* /. `'{"BucketFolder": " sourceData "}'`
`BucketName`	Le nom du compartiment S3. `'{"BucketName": " myBucket"}'`
`CdcPath`	Emplacement des fichiers CDC. Cet attribut est requis si une tâche capture les données modifiées ; dans le cas contraire, il est facultatif. Si `CdcPath` est présent, AWS DMS lit les fichiers CDC à partir de ce chemin et réplique les modifications des données au point de terminais on cible. Pour de plus amples informations, veuillez consulter Utilisation de la CDC avec Amazon S3 en tant que source pour AWS DMS. `'{"CdcPath": " changeData "}'`
`CsvDelimiter`	Le délimiteur utilisé pour séparer les colonnes des fichiers sources. La valeur par défaut est une virgule. Un exemple suit. `'{"CsvDelimiter": ","}'`
`CsvNullValue`	Chaîne définie par l'utilisateur traitée par AWS DMS comme nulle lors de la lecture depuis la source. La valeur par défaut est une chaîne vide. Si vous ne définissez pas ce paramètre, AWS DMS traite une chaîne vide comme une valeur nulle. Si vous définissez ce paramètre sur une chaîne telle que « \N », AWS DMS traite cette chaîne comme une valeur nulle et traite les chaînes vides comme une valeur de chaîne vide.
`CsvRowDelimiter`	Le délimiteur utilisé pour séparer les lignes des fichiers sources. La valeur par défaut est un caractère de saut de ligne (\n). `'{"CsvRowDelimiter": "\n"}'`

Option	Description
`IgnoreHeaderRows`	Lorsque cette valeur est définie sur 1, AWS DMS ignore la première ligne d'en-tête dans un fichier .csv. La valeur 1 active la fonctionnalité, la valeur 0 désactive la fonctionnalité. La valeur par défaut est 0. `'{"IgnoreHeaderRows": 1}'`
`Rfc4180`	Lorsque cette valeur est définie sur `true` ou `y`, chaque guillemet double de tête doit être suivi d'un guillemet double de fin. Ce format est conforme à RFC 4180. Lorsque ce paramètre est défini sur `false` ou `n`, les chaînes littérales sont copiées vers la cible en l'état. Dans ce cas, un délimiteur de colonne ou de ligne indique la fin du champ. Par conséquent, vous ne pouvez pas utiliser un délimiteur dans le cadre de la chaîne, car il indique une fin de la valeur. La valeur par défaut est `true`. Valeurs valides : `true`, `false`, `y`, `n` `'{"Rfc4180": false}'`

Types de données source pour Amazon S3

La migration de données qui utilise Amazon S3 en tant que source pour AWS DMS doit mapper les données d'Amazon S3 aux types de données AWS DMS. Pour de plus amples informations, veuillez consulter Définition de tables externes pour Amazon S3 en tant que source pour AWS DMS.

Pour en savoir plus sur la façon d'afficher le type de données qui est mappé dans la cible, consultez la section concernant le point de terminaison cible que vous utilisez.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Les types de données AWS DMS suivants sont utilisés avec Amazon S3 en tant que source :

- BYTE : requiert `ColumnLength`. Pour de plus amples informations, veuillez consulter Définition de tables externes pour Amazon S3 en tant que source pour AWS DMS.

- DATE
- TIME
- DATETIME : pour plus d'informations et pour obtenir un exemple, consultez l'exemple de type DATETIME dans Définition de tables externes pour Amazon S3 en tant que source pour AWS DMS.
- INT1
- INT2
- INT4
- INT8
- NUMERIC : requiert `ColumnPrecision` et `ColumnScale`. AWS DMS prend en charge les valeurs maximales suivantes :
  - ColumnPrecision : 38
  - ColumnScale : 31

  Pour plus d'informations et pour obtenir un exemple, consultez l'exemple de type NUMERIC dans Définition de tables externes pour Amazon S3 en tant que source pour AWS DMS.
- REAL4
- REAL8
- STRING : requiert `ColumnLength`. Pour de plus amples informations, veuillez consulter Définition de tables externes pour Amazon S3 en tant que source pour AWS DMS.
- UINT1
- UINT2
- UINT4
- UINT8
- BLOB
- CLOB
- BOOLEAN

Utilisation d'IBM Db2 pour Linux, Unix, Windows et de la base de données Amazon RDS (Db2 LUW) comme source pour AWS DMS

Vous pouvez migrer les données d'une base de données IBM Db2 pour Linux, Unix, Windows et Amazon RDS (Db2 LUW) vers n'importe quelle base de données cible prise en charge à l'aide de (). AWS Database Migration Service AWS DMS

Pour plus d'informations sur les versions de Db2 sous Linux, Unix, Windows et RDS prises AWS DMS en charge en tant que source, consultez. Sources pour AWS DMS

Vous pouvez utiliser le protocole SSL pour chiffrer les connexions entre votre point de terminaison Db2 LUW et l'instance de réplication. Pour plus d'informations sur l'utilisation de SSL avec un point de terminaison Db2 LUW, consultez Utilisation du protocole SSL avec AWS Database Migration Service.

Conditions préalables à l'utilisation de DB2 LUW comme source pour AWS DMS

Les conditions préalables suivantes sont requises pour que vous puissiez utiliser une base de données Db2 LUW comme source.

Pour activer la réplication continue, également appelée capture de données modifiées (CDC), effectuez les opérations suivantes :

- Définissez la base de données pour qu'elle soit récupérable, ce qui AWS DMS nécessite de capturer les modifications. Une base de données est récupérable si l'un ou les deux paramètres de configuration de base de données `LOGARCHMETH1` ou `LOGARCHMETH2` sont définis sur `ON`.

  Si votre base de données est récupérable, vous AWS DMS pouvez accéder au Db2 `ARCHIVE LOG` si nécessaire.
- Assurez-vous que les journaux de transactions DB2 sont disponibles, avec une période de conservation suffisante pour être traités par AWS DMS.
- DB2 a besoin d'une autorisation `SYSADM` ou `DBADM` pour extraire les enregistrements des journaux de transactions. Accordez les autorisations suivantes au compte utilisateur :
  - `SYSADM` ou `DBADM`
  - `DATAACCESS`

> **Note**
>
> Pour les tâches à chargement complet uniquement, le compte d'utilisateur DMS doit disposer de l'autorisation DATAACCESS.

- Lorsque vous utilisez IBM DB2 for LUW version 9.7 en tant que source, définissez l'attribut de connexion supplémentaire (ECA), `CurrentLSN`, comme suit :

  `CurrentLSN=`*`LSN`*, où *`LSN`* spécifie un numéro de séquence de journal (LSN) à partir duquel vous souhaitez démarrer la réplication. Ou `CurrentLSN=`*`scan`*.

Limitations liées à l'utilisation de DB2 LUW comme source pour AWS DMS

AWS DMS ne prend pas en charge les bases de données en cluster. Toutefois, vous pouvez définir une base de données Db2 LUW distincte pour chacun des points de terminaison d'un cluster. Par exemple, vous pouvez créer une tâche de migration à chargement complet avec l'un des nœuds du cluster, puis créer des tâches distinctes à partir de chaque nœud.

AWS DMS ne prend pas en charge le type de `BOOLEAN` données de votre base de données source DB2 LUW.

Les limitations suivantes s'appliquent lorsque vous utilisez la réplication continue (CDC) :

- Lorsqu'une table comportant plusieurs partitions est tronquée, le nombre d'événements DDL affichés dans la AWS DMS console est égal au nombre de partitions. En effet, Db2 LUW enregistre une DDL distincte pour chaque partition.
- Les actions de DDL suivantes ne sont pas prises en charge sur les tables partitionnées :
  - ALTER TABLE ADD PARTITION
  - ALTER TABLE DETACH PARTITION
  - ALTER TABLE ATTACH PARTITION
- AWS DMS ne prend pas en charge une migration de réplication continue depuis une instance de secours DB2 High Availability Disaster Recovery (HADR). L'instance de secours n'est pas accessible.
- Le type de données DECFLOAT n'est pas pris en charge. Par conséquent, les modifications apportées aux colonnes DECFLOAT sont ignorées lors de la réplication continue.
- L'instruction RENAME COLUMN n'est pas prise en charge.

- Lorsque vous effectuez des mises à jour de tables de clustering multidimensionnel (MDC), chaque mise à jour est affichée dans la AWS DMS console sous la forme INSERT + DELETE.
- Lorsque le paramètre de tâche Inclure les colonnes LOB dans la réplication n'est pas activé, toute table comportant des colonnes LOB est suspendue pendant la réplication continue.
- Pour les versions 10.5 et supérieures de DB2 LUW, les colonnes de chaîne de longueur variable contenant des données stockées sont ignorées. out-of-row Cette limitation s'applique uniquement aux tables créées avec une taille de ligne étendue pour les colonnes dont les types de données sont VARCHAR et VARGRAPHIC. Pour contourner cette limitation, déplacez la table vers un espace de table dont la taille de page est supérieure. Pour plus d'informations, consultez What can I do if I want to change the pagesize of DB2 tablespaces.
- Pour une réplication continue, DMS ne prend pas en charge la migration des données chargées au niveau de la page par l'utilitaire DB2 LOAD. Utilisez plutôt l'utilitaire IMPORT qui utilise des insertions SQL. Pour plus d'informations, consultez les différences entre les utilitaires d'importation et de chargement.
- Pendant l'exécution d'une tâche de réplication, DMS capture les DDL CREATE TABLE uniquement si les tables ont été créées avec l'attribut DATA CAPTURE CHANGE.
- DMS présente les limites suivantes lors de l'utilisation de la fonctionnalité de partition de base de données (DPF) DB2 :
  - DMS ne peut pas coordonner les transactions entre les nœuds DB2 dans un environnement DPF. Cela est dû à des contraintes au sein de l'interface API IBM DB2READLOG. Dans le format DPF, les transactions peuvent s'étendre sur plusieurs nœuds DB2, en fonction de la manière dont DB2 partitionne les données. Par conséquent, votre solution DMS doit capturer les transactions de chaque nœud DB2 indépendamment.
  - DMS peut capturer les transactions locales à partir de chaque nœud Db2 du cluster DPF en le configurant 1 sur plusieurs points de `connectNode` terminaison sources DMS. Cette configuration correspond aux numéros de nœuds logiques définis dans le fichier `db2nodes.cfg` de configuration du serveur DB2.
  - Les transactions locales sur des nœuds Db2 individuels peuvent faire partie d'une transaction globale plus importante. DMS applique chaque transaction locale indépendamment sur la cible, sans coordination avec les transactions sur les autres nœuds Db2. Ce traitement indépendant peut entraîner des complications, notamment lorsque des lignes sont déplacées entre des partitions.
  - Lorsque DMS se réplique à partir de plusieurs nœuds Db2, il n'y a aucune garantie quant à l'ordre correct des opérations sur la cible, car le DMS applique les opérations indépendamment

pour chaque nœud Db2. Vous devez vous assurer que la capture des transactions locales indépendamment de chaque nœud DB2 fonctionne pour votre cas d'utilisation spécifique.

- Lors de la migration depuis un environnement DPF, nous vous recommandons d'exécuter d'abord une tâche de chargement complet sans événements mis en cache, puis d'exécuter des tâches uniquement en mode CDC. Nous recommandons d'exécuter une tâche par nœud DB2, en commençant par l'horodatage de début du chargement complet ou le LRI (identifiant d'enregistrement du journal) que vous avez défini à l'aide du paramètre du point de terminaison. `StartFromContext` Pour plus d'informations sur la détermination du point de départ de la réplication, consultez la section Recherche de la valeur LSN ou LRI pour le démarrage de la réplication dans la documentation IBM Support.

- Pour la réplication continue (CDC), si vous prévoyez de démarrer la réplication à partir d'un horodatage spécifique, vous devez définir l'attribut de connexion `StartFromContext` sur l'horodatage requis.
- À l'heure actuelle, DMS ne prend pas en charge la fonctionnalité Db2 pureScale, une extension de DB2 LUW que vous pouvez utiliser pour mettre à l'échelle votre solution de base de données.
- AWS DMS ne prend pas en charge le CDC lors de l'utilisation de Db2 pour Amazon RDS comme source.

Paramètres du point de terminaison lors de l'utilisation de DB2 LUW comme source pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données source Db2 LUW comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison source à l'aide de la AWS DMS console ou à l'aide de la `create-endpoint` commande dans le AWS CLI, avec la syntaxe `--ibm-db2-settings '{"EndpointSetting": "value", ...}'` JSON.

Les paramètres de point de terminaison que vous pouvez utiliser avec Db2 LUW en tant que source sont indiqués dans le tableau suivant.

Name (Nom)	Description
`CurrentLSN`	Pour une réplication en continu (CDC), utilisez `CurrentLSN` pour spécifier un numéro de séquence de journal où vous voulez que la réplication démarre.

Name (Nom)	Description
`MaxKBytesPerRead`	Nombre maximal d'octets par lecture, sous forme d'une valeur NUMÉRIQUE (NUMBER). La valeur par défaut est de 64 Ko.
`SetDataCaptureChanges`	Active la réplication en continu (CDC) en tant que valeur BOOLÉENNE. Par défaut, la valeur est true.

Name (Nom)	Description

`StartFromContext`	Pour une réplication continue (CDC), utilisez `StartFromContext` pour spécifier la limite inférieure d'un journal à partir de laquelle démarrer la réplication. `StartFromContext` accepte différentes formes de valeurs. Les valeurs valides sont les suivantes :
	• `timestamp` (UTC). Par exemple :
	`'{"StartFromContext": "timestamp:2021-09-21T13:00:00"}'`
	• `NOW`
	Pour IBM DB2 LUW versions 10.5 et ultérieures, NOW combiné à CurrentLSN: scan démarre la tâche à partir du LSO le plus récent. Par exemple :
	`'{"CurrentLSN": "scan", "StartFromContext": "NOW"}'`
	• Un LRI spécifique. Par exemple :
	`'{"StartFromContext": "0100000000000022C000000000004FB13"}'`
	Pour déterminer la plage LRI/LSN d'un fichier journal, exécutez la commande `db2flsn` comme indiqué dans l'exemple suivant.
	`db2flsn -db SAMPLE -lrirange 2`
	La sortie est similaire à l'exemple suivant.

Name (Nom)	Description

	`S0000002.LOG: has LRI range 00000000000000100 00000000002254000000000004F9A6 to 00000000000000010000000000022CC000000000004 FB13`
	
	Dans cette sortie, le fichier journal est S0000002.LOG et la valeur du StartFromContextLRI correspond aux 34 octets situés à la fin de la plage.
	
	`0100000000000022CC000000000004FB13`

Types de données source pour IBM Db2 LUW

La migration de données qui utilise Db2 LUW comme source prend en AWS DMS charge la plupart des types de données DB2 LUW. Le tableau suivant indique les types de données source DB2 LUW pris en charge lors de l'utilisation AWS DMS et le mappage par défaut à partir AWS DMS des types de données. Pour plus d'informations sur les types de donnée Db2 LUW, consultez la documentation Db2 LUW.

Pour obtenir des informations sur la façon d'afficher le type de données qui est mappé dans la cible, consultez la section relative au point de terminaison cible que vous utilisez.

Pour plus d'informations sur AWS DMS les types de données, consultezTypes de données pour AWS Database Migration Service.

Types de données Db2 LUW	AWS DMS types de données
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
DECIMAL (p,s)	NUMERIC (p,s)
FLOAT	REAL8

Types de données Db2 LUW	AWS DMS types de données
DOUBLE	REAL8
REAL	REAL4
DECFLOAT (p)	Si la précision est de 16, alors REAL8 ; si la précision est 34, alors STRING
GRAPHIC (n)	WSTRING, pour des chaînes graphiques à longueur fixe de caractères à deux octets, avec une longueur supérieure à 0 et inférieure ou égale à 127
VARGRAPHIC (n)	WSTRING, pour des chaînes graphiques à longueur variable, avec une longueur supérieur e à 0 et inférieure ou égale à 16 352 caractères à 2 octets
LONG VARGRAPHIC (n)	CLOB, pour des chaînes graphiques à longueur variable, avec une longueur supérieur e à 0 et inférieure ou égale à 16 352 caractères à 2 octets
CHARACTER (n)	STRING, pour des chaînes à longueur fixe de caractères à deux octets, avec une longueur supérieure à 0 et inférieure ou égale à 255
VARCHAR(n)	STRING, pour des chaînes à longueur variable de caractères à deux octets, avec une longueur supérieure à 0 et inférieure ou égale à 32,704
LONG VARCHAR (n)	CLOB, pour des chaînes à longueur variable de caractères à deux octets, avec une longueur supérieure à 0 et inférieure ou égale à 32,704
CHAR (n) FOR BIT DATA	BYTES
VARCHAR (n) FOR BIT DATA	BYTES

Types de données Db2 LUW	AWS DMS types de données
LONG VARCHAR FOR BIT DATA	BYTES
DATE	DATE
TIME	TIME
TIMESTAMP	DATETIME
BLOB (n)	BLOB La longueur maximale est de 2 147 483 647 octets
CLOB (n)	CLOB La longueur maximale est de 2 147 483 647 octets
DBCLOB (n)	CLOB La longueur maximale est de 1 073 741 824 caractères à deux octets
xml	CLOB

Utilisation de bases de données IBM Db2 for z/OS en tant que source pour AWS DMS

Vous pouvez migrer des données à partir d'une base de données IBM for z/OS vers n'importe quelle base de données cible prise en charge à l'aide d'AWS Database Migration Service (AWS DMS).

Pour en savoir plus sur les versions de Db2 for z/OS qui sont prises en charge par AWS DMS en tant que source, consultez Sources pour AWS DMS.

Conditions préalables à l'utilisation de Db2 for z/OS en tant que source pour AWS DMS

Pour utiliser une base de données IBM Db2 for z/OS en tant que source dans AWS DMS, accordez les privilèges suivants à l'utilisateur Db2 for z/OS spécifié dans les paramètres de connexion au point de terminaison source.

```
GRANT SELECT ON SYSIBM.SYSTABLES TO Db2USER;
GRANT SELECT ON SYSIBM.SYSTABLESPACE TO Db2USER;
GRANT SELECT ON SYSIBM.SYSTABLEPART TO Db2USER;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO Db2USER;
GRANT SELECT ON SYSIBM.SYSDATABASE TO Db2USER;
GRANT SELECT ON SYSIBM.SYSDUMMY1 TO Db2USER
```

Accordez également les tables source SELECT ON *user defined*.

Un point de terminaison source IBM Db2 for z/OS AWS DMS repose sur IBM Data Server Driver for ODBC pour accéder aux données. Le serveur de base de données doit disposer d'une licence IBM ODBC Connect valide pour que DMS puisse se connecter à ce point de terminaison.

Limitations relatives à l'utilisation de Db2 for z/OS en tant que source pour AWS DMS

Les limitations suivantes s'appliquent lors de l'utilisation d'une base de données IBM Db2 for z/OS en tant que source pour AWS DMS :

- Seules les tâches de réplication à chargement complet sont prises en charge. La capture des données modifiées (CDC) n'est pas prise en charge.
- Le chargement parallèle n'est pas pris en charge.
- La validation de données des vues n'est pas prise en charge.
- Les noms de schéma, de table et de colonne doivent être spécifiés en MAJUSCULES dans les mappages de tables pour les transformations au niveau des colonnes/tables et les filtres de sélection au niveau des lignes.

Types de données source pour IBM Db2 for z/OS

Les migrations de données qui utilisent Db2 for z/OS en tant que source pour AWS DMS prennent en charge la plupart des types de données Db2 for z/OS. Le tableau suivant présente les types de

données source Db2 for z/OS qui sont pris en charge lorsque vous utilisez AWS DMS et le mappage par défaut des types de données AWS DMS.

Pour plus d'informations sur les types de donnée Db2 for z/OS, consultez la documentation d'IBM Db2 for z/OS.

Pour obtenir des informations sur la façon d'afficher le type de données qui est mappé dans la cible, consultez la section relative au point de terminaison cible que vous utilisez.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Types de données Db2 for z/OS	Types de données AWS DMS
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
DECIMAL (p,s)	NUMERIC (p,s) Si la virgule décimale est définie sur la virgule (,) dans la configuration DB2, configurez Replicate pour prendre en charge le paramètre DB2.
FLOAT	REAL8
DOUBLE	REAL8
REAL	REAL4
DECFLOAT (p)	Si la précision est de 16, alors REAL8 ; si la précision est 34, alors STRING
GRAPHIC (n)	Si n>=127, alors WSTRING, pour des chaînes graphiques à longueur fixe de caractères à deux octets, avec une longueur supérieure à 0 et inférieure ou égale à 127

Types de données Db2 for z/OS	Types de données AWS DMS
VARGRAPHIC (n)	WSTRING, pour des chaînes graphiques à longueur variable, avec une longueur supérieure à 0 et inférieure ou égale à 16 352 caractères à 2 octets
LONG VARGRAPHIC (n)	CLOB, pour des chaînes graphiques à longueur variable, avec une longueur supérieure à 0 et inférieure ou égale à 16 352 caractères à 2 octets
CHARACTER (n)	STRING, pour des chaînes à longueur fixe de caractères à deux octets, avec une longueur supérieure à 0 et inférieure ou égale à 255
VARCHAR(n)	STRING, pour des chaînes à longueur variable de caractères à deux octets, avec une longueur supérieure à 0 et inférieure ou égale à 32,704
LONG VARCHAR (n)	CLOB, pour des chaînes à longueur variable de caractères à deux octets, avec une longueur supérieure à 0 et inférieure ou égale à 32,704
CHAR (n) FOR BIT DATA	BYTES
VARCHAR (n) FOR BIT DATA	BYTES
LONG VARCHAR FOR BIT DATA	BYTES
DATE	DATE
TIME	TIME
TIMESTAMP	DATETIME
BLOB (n)	BLOB La longueur maximale est de 2 147 483 647 octets

Types de données Db2 for z/OS	Types de données AWS DMS
CLOB (n)	CLOB La longueur maximale est de 2 147 483 647 octets
DBCLOB (n)	CLOB La longueur maximale est de 1 073 741 824 caractères à deux octets
xml	CLOB
BINAIRE	BYTES
VARBINARY	BYTES
ROWID	BYTES. Pour plus d'informations sur l'utilisation de ROWID, consultez les rubriques suivantes.
TIMESTAMP WITH TIME ZONE	Non pris en charge.

Les colonnes ROWID sont migrées par défaut lorsque le mode de préparation des tables cibles pour la tâche est défini sur DROP_AND_CREATE (par défaut). La validation des données ignore ces colonnes, car les lignes n'ont aucun sens en dehors de la base de données et de la table spécifiques. Pour désactiver la migration de ces colonnes, vous pouvez effectuer l'une des étapes préparatoires suivantes :

- Pré-créez la table cible sans ces colonnes. Définissez ensuite le mode de préparation des tables cibles de la tâche sur DO_NOTHING ou TRUNCATE_BEFORE_LOAD. Vous pouvez utiliser AWS Schema Conversion Tool (AWS SCT) pour pré-créer la table cible sans les colonnes.
- Ajoutez une règle de mappage de table à une tâche qui filtre ces colonnes afin qu'elles soient ignorées. Pour de plus amples informations, veuillez consulter Règles et actions de transformation.

Classements EBCDIC dans PostgreSQL pour le service AWS Mainframe Modernization

Le programme AWS Mainframe Modernization vous aide à moderniser vos applications mainframe pour les adapter aux environnements d'exécution gérés par AWS. Il fournit des outils et des ressources pour vous aider à planifier et à implémenter vos projets de migration et de modernisation. Pour plus d'informations sur la modernisation et la migration d'une application mainframe, consultez Modernisation de mainframe avec AWS.

Certains jeux de données IBM Db2 for z/OS sont codés avec le jeu de caractères EBCDIC (Extended Binary Coded Decimal Interchange). Ce jeu de caractères développé avant ASCII (American Standard Code for Information Interchange) est désormais couramment utilisé. Une page de code mappe chaque caractère du texte aux caractères d'un jeu de caractères. Une page de code traditionnelle contient les informations de mappage entre un point de code et un ID de caractère. Un ID de caractère est une chaîne de données de caractères de 8 octets. Un point de code est un nombre binaire de 8 bits qui représente un caractère. Les points de code sont généralement des représentations hexadécimales de leurs valeurs binaires.

Si vous utilisez actuellement le composant Micro Focus ou BluAge du service Mainframe Modernization, vous devez demander à AWS DMS de décaler (traduire) certains points de code. Vous pouvez utiliser les paramètres de tâche AWS DMS pour procéder aux décalages. L'exemple suivant illustre comment utiliser l'opération AWS DMS `CharacterSetSettings` pour mapper les décalages dans un paramètre de tâche DMS.

```
"CharacterSetSettings": {
        "CharacterSetSupport": null,
        "CharacterReplacements": [
{"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0178"}
            }
        ]
    }
```

Certains classements EBCDIC existent déjà pour PostgreSQL et comprennent le décalage nécessaire. Plusieurs pages de code différentes sont prises en charge. Les sections suivantes fournissent des exemples JSON de ce que vous devez décaler pour toutes les pages de code prises en charge. Vous pouvez simplement copier/coller le code JSON dont vous avez besoin dans votre tâche DMS.

Classements EBCDIC spécifiques à Micro Focus

Pour Micro Focus, décalez un sous-ensemble de caractères selon vos besoins pour les classements suivants.

```
da-DK-cp1142m-x-icu
de-DE-cp1141m-x-icu
en-GB-cp1146m-x-icu
en-US-cp1140m-x-icu
es-ES-cp1145m-x-icu
fi-FI-cp1143m-x-icu
fr-FR-cp1147m-x-icu
it-IT-cp1144m-x-icu
nl-BE-cp1148m-x-icu
```

Example Décalages de données Micro Focus par classement :

en_us_cp1140m

Décalage de code :

```
0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0161"}
```

```
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0178"}
```

en_us_cp1141m

Décalage de code :

```
0000    0180
00B8    0160
00BC    0161
00BD    017D
00BE    017E
00A8    0152
00B4    0153
00A6    0178
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0178"}
```

en_us_cp1142m

Décalage de code :

```
0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
```

```
00A8    0153
00B4    0178
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0178"}
```

en_us_cp1143m

Décalage de code :

```
0000    0180
00B8    0160
00BC    0161
00BD    017D
00BE    017E
00A8    0152
00B4    0153
00A6    0178
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0178"}
```

en_us_cp1144m

Décalage de code :

```
0000    0180
00B8    0160
00BC    0161
00BD    017D
00BE    017E
00A8    0152
00B4    0153
00A6    0178
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0178"}
```

en_us_cp1145m

Décalage de code :

```
0000    0180
00A6    0160
00B8    0161
00A8    017D
00BC    017E
00BD    0152
00BE    0153
00B4    0178
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0178"}
```

en_us_cp1146m

Décalage de code :

```
0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0178"}
```

en_us_cp1147m

Décalage de code :

```
0000    0180
```

```
00B8    0160
00A8    0161
00BC    017D
00BD    017E
00BE    0152
00B4    0153
00A6    0178
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0178"}
```

en_us_cp1148m

Décalage de code :

```
0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "0152"}
```

```
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0178"}
```

Classements EBCDIC spécifiques à BluAge

Pour BluAge, décalez toutes les valeurs basses et valeurs hautes suivantes selon vos besoins. Ces classements ne doivent être utilisés que pour prendre en charge le service Mainframe Migration BluAge.

```
da-DK-cp1142b-x-icu
 da-DK-cp277b-x-icu
 de-DE-cp1141b-x-icu
 de-DE-cp273b-x-icu
 en-GB-cp1146b-x-icu
 en-GB-cp285b-x-icu
 en-US-cp037b-x-icu
 en-US-cp1140b-x-icu
 es-ES-cp1145b-x-icu
 es-ES-cp284b-x-icu
 fi-FI-cp1143b-x-icu
 fi-FI-cp278b-x-icu
 fr-FR-cp1147b-x-icu
 fr-FR-cp297b-x-icu
 it-IT-cp1144b-x-icu
 it-IT-cp280b-x-icu
 nl-BE-cp1148b-x-icu
 nl-BE-cp500b-x-icu
```

Example Décalages de données BluAge :

da-DK-cp277b et da-DK-cp1142b

Décalage de code :

```
0180    0180
0001    0181
0002    0182
0003    0183
009C    0184
0009    0185
0086    0186
```

```
007F    0187
0097    0188
008D    0189
008E    018A
000B    018B
000C    018C
000D    018D
000E    018E
000F    018F
0010    0190
0011    0191
0012    0192
0013    0193
009D    0194
0085    0195
0008    0196
0087    0197
0018    0198
0019    0199
0092    019A
008F    019B
001C    019C
001D    019D
001E    019E
001F    019F
0080    01A0
0081    01A1
0082    01A2
0083    01A3
0084    01A4
000A    01A5
0017    01A6
001B    01A7
0088    01A8
0089    01A9
008A    01AA
008B    01AB
008C    01AC
0005    01AD
0006    01AE
0007    01AF
0090    01B0
0091    01B1
0016    01B2
```

```
0093    01B3
0094    01B4
0095    01B5
0096    01B6
0004    01B7
0098    01B8
0099    01B9
009A    01BA
009B    01BB
0014    01BC
0015    01BD
009E    01BE
001A    01BF
009F    027F
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
```

```
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

de-DE-273b et de-DE-1141b

Décalage de code :

```
0180    0180
0001    0181
0002    0182
0003    0183
009C    0184
0009    0185
0086    0186
007F    0187
0097    0188
008D    0189
008E    018A
000B    018B
000C    018C
000D    018D
000E    018E
000F    018F
0010    0190
0011    0191
0012    0192
0013    0193
009D    0194
0085    0195
0008    0196
0087    0197
0018    0198
0019    0199
0092    019A
008F    019B
001C    019C
001D    019D
001E    019E
001F    019F
0080    01A0
0081    01A1
0082    01A2
0083    01A3
0084    01A4
000A    01A5
0017    01A6
001B    01A7
```

```
0088    01A8
0089    01A9
008A    01AA
008B    01AB
008C    01AC
0005    01AD
0006    01AE
0007    01AF
0090    01B0
0091    01B1
0016    01B2
0093    01B3
0094    01B4
0095    01B5
0096    01B6
0004    01B7
0098    01B8
0099    01B9
009A    01BA
009B    01BB
0014    01BC
0015    01BD
009E    01BE
001A    01BF
009F    027F
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
```

```
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
```

```
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

en-GB-285b et en-GB-1146b

Décalage de code :

```
0180    0180
0001    0181
0002    0182
0003    0183
009C    0184
0009    0185
0086    0186
007F    0187
0097    0188
008D    0189
008E    018A
000B    018B
000C    018C
000D    018D
000E    018E
000F    018F
0010    0190
0011    0191
0012    0192
0013    0193
009D    0194
0085    0195
0008    0196
0087    0197
0018    0198
0019    0199
0092    019A
008F    019B
001C    019C
001D    019D
```

```
001E    019E
001F    019F
0080    01A0
0081    01A1
0082    01A2
0083    01A3
0084    01A4
000A    01A5
0017    01A6
001B    01A7
0088    01A8
0089    01A9
008A    01AA
008B    01AB
008C    01AC
0005    01AD
0006    01AE
0007    01AF
0090    01B0
0091    01B1
0016    01B2
0093    01B3
0094    01B4
0095    01B5
0096    01B6
0004    01B7
0098    01B8
0099    01B9
009A    01BA
009B    01BB
0014    01BC
0015    01BD
009E    01BE
001A    01BF
009F    027F
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
{"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
```

```
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
```

```
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

en-us-037b et en-us-1140b

Décalage de code :

```
0180    0180
0001    0181
0002    0182
0003    0183
009C    0184
0009    0185
0086    0186
007F    0187
0097    0188
008D    0189
008E    018A
000B    018B
000C    018C
000D    018D
000E    018E
000F    018F
0010    0190
0011    0191
0012    0192
0013    0193
```

```
009D    0194
0085    0195
0008    0196
0087    0197
0018    0198
0019    0199
0092    019A
008F    019B
001C    019C
001D    019D
001E    019E
001F    019F
0080    01A0
0081    01A1
0082    01A2
0083    01A3
0084    01A4
000A    01A5
0017    01A6
001B    01A7
0088    01A8
0089    01A9
008A    01AA
008B    01AB
008C    01AC
0005    01AD
0006    01AE
0007    01AF
0090    01B0
0091    01B1
0016    01B2
0093    01B3
0094    01B4
0095    01B5
0096    01B6
0004    01B7
0098    01B8
0099    01B9
009A    01BA
009B    01BB
0014    01BC
0015    01BD
009E    01BE
001A    01BF
```

```
009F    027F
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
{"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
```

```
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

es-ES-284b et es-ES-1145b

Décalage de code :

```
0180    0180
0001    0181
0002    0182
0003    0183
009C    0184
0009    0185
0086    0186
007F    0187
0097    0188
008D    0189
```

```
008E    018A
000B    018B
000C    018C
000D    018D
000E    018E
000F    018F
0010    0190
0011    0191
0012    0192
0013    0193
009D    0194
0085    0195
0008    0196
0087    0197
0018    0198
0019    0199
0092    019A
008F    019B
001C    019C
001D    019D
001E    019E
001F    019F
0080    01A0
0081    01A1
0082    01A2
0083    01A3
0084    01A4
000A    01A5
0017    01A6
001B    01A7
0088    01A8
0089    01A9
008A    01AA
008B    01AB
008C    01AC
0005    01AD
0006    01AE
0007    01AF
0090    01B0
0091    01B1
0016    01B2
0093    01B3
0094    01B4
0095    01B5
```

```
0096    01B6
0004    01B7
0098    01B8
0099    01B9
009A    01BA
009B    01BB
0014    01BC
0015    01BD
009E    01BE
001A    01BF
009F    027F
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
```

```
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

fi_FI-278b et fi-FI-1143b

Décalage de code :

```
0180    0180
0001    0181
0002    0182
0003    0183
009C    0184
0009    0185
0086    0186
007F    0187
0097    0188
008D    0189
008E    018A
000B    018B
000C    018C
000D    018D
000E    018E
000F    018F
0010    0190
0011    0191
0012    0192
0013    0193
009D    0194
0085    0195
0008    0196
0087    0197
0018    0198
0019    0199
0092    019A
008F    019B
001C    019C
001D    019D
001E    019E
001F    019F
0080    01A0
0081    01A1
0082    01A2
0083    01A3
0084    01A4
000A    01A5
0017    01A6
001B    01A7
0088    01A8
0089    01A9
008A    01AA
```

```
008B    01AB
008C    01AC
0005    01AD
0006    01AE
0007    01AF
0090    01B0
0091    01B1
0016    01B2
0093    01B3
0094    01B4
0095    01B5
0096    01B6
0004    01B7
0098    01B8
0099    01B9
009A    01BA
009B    01BB
0014    01BC
0015    01BD
009E    01BE
001A    01BF
009F    027F
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
```

```
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
```

```
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

fr-FR-297b et fr-FR-1147b

Décalage de code :

```
0180    0180
0001    0181
0002    0182
0003    0183
009C    0184
0009    0185
0086    0186
007F    0187
0097    0188
008D    0189
008E    018A
000B    018B
000C    018C
000D    018D
000E    018E
000F    018F
0010    0190
0011    0191
0012    0192
0013    0193
009D    0194
0085    0195
0008    0196
0087    0197
0018    0198
0019    0199
0092    019A
008F    019B
001C    019C
001D    019D
001E    019E
001F    019F
0080    01A0
0081    01A1
```

```
0082    01A2
0083    01A3
0084    01A4
000A    01A5
0017    01A6
001B    01A7
0088    01A8
0089    01A9
008A    01AA
008B    01AB
008C    01AC
0005    01AD
0006    01AE
0007    01AF
0090    01B0
0091    01B1
0016    01B2
0093    01B3
0094    01B4
0095    01B5
0096    01B6
0004    01B7
0098    01B8
0099    01B9
009A    01BA
009B    01BB
0014    01BC
0015    01BD
009E    01BE
001A    01BF
009F    027F
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
{"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
```

```
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
```

```
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

it-IT-280b et it-IT-1144b

Décalage de code :

```
0180    0180
0001    0181
0002    0182
0003    0183
009C    0184
0009    0185
0086    0186
007F    0187
0097    0188
008D    0189
008E    018A
000B    018B
000C    018C
000D    018D
000E    018E
000F    018F
0010    0190
0011    0191
0012    0192
0013    0193
009D    0194
0085    0195
0008    0196
0087    0197
```

```
0018    0198
0019    0199
0092    019A
008F    019B
001C    019C
001D    019D
001E    019E
001F    019F
0080    01A0
0081    01A1
0082    01A2
0083    01A3
0084    01A4
000A    01A5
0017    01A6
001B    01A7
0088    01A8
0089    01A9
008A    01AA
008B    01AB
008C    01AC
0005    01AD
0006    01AE
0007    01AF
0090    01B0
0091    01B1
0016    01B2
0093    01B3
0094    01B4
0095    01B5
0096    01B6
0004    01B7
0098    01B8
0099    01B9
009A    01BA
009B    01BB
0014    01BC
0015    01BD
009E    01BE
001A    01BF
009F    027F
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
```

```
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

nl-BE-500b et nl-BE-1148b

Décalage de code :

```
0180    0180
0001    0181
0002    0182
0003    0183
009C    0184
0009    0185
0086    0186
007F    0187
0097    0188
008D    0189
008E    018A
000B    018B
000C    018C
000D    018D
```

```
000E    018E
000F    018F
0010    0190
0011    0191
0012    0192
0013    0193
009D    0194
0085    0195
0008    0196
0087    0197
0018    0198
0019    0199
0092    019A
008F    019B
001C    019C
001D    019D
001E    019E
001F    019F
0080    01A0
0081    01A1
0082    01A2
0083    01A3
0084    01A4
000A    01A5
0017    01A6
001B    01A7
0088    01A8
0089    01A9
008A    01AA
008B    01AB
008C    01AC
0005    01AD
0006    01AE
0007    01AF
0090    01B0
0091    01B1
0016    01B2
0093    01B3
0094    01B4
0095    01B5
0096    01B6
0004    01B7
0098    01B8
0099    01B9
```

```
009A    01BA
009B    01BB
0014    01BC
0015    01BD
009E    01BE
001A    01BF
009F    027F
```

Mappage d'entrée correspondant pour une tâche AWS DMS :

```
 {"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
```

```
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

Cibles pour la migration des données

AWS Database Migration Service (AWS DMS) peut utiliser plusieurs des bases de données les plus répandues comme cible pour la réplication de données. La cible peut être une instance Amazon Elastic Compute Cloud (Amazon EC2), une instance Amazon Relational Database Service (Amazon RDS) ou une base de données sur site.

Pour une liste complète des cibles valides, consultez Cibles pour AWS DMS.

> **Note**
>
> AWS DMS ne prend pas en charge la migration entre les régions AWS pour les types de point de terminaison cible suivants :
>
> - Amazon DynamoDB
> - Amazon OpenSearch Service
> - Amazon Kinesis Data Streams

Rubriques

- Utilisation d'une base de données Oracle comme cible pour AWS Database Migration Service
- Utilisation d'une base de données Microsoft SQL Server comme cible pour AWS Database Migration Service
- Utilisation d'une base de données PostgreSQL comme cible pour AWS Database Migration Service
- Utilisation d'une base de données compatible MySQL comme cible pour AWS Database Migration Service
- Utilisation d'une base de données Amazon Redshift en tant que cible pour AWS Database Migration Service
- Utilisation d'une base de données SAP ASE comme cible pour AWS Database Migration Service
- Utilisation d'Amazon S3 en tant que cible pour AWS Database Migration Service
- Utilisation d'une base de données Amazon DynamoDB en tant que cible pour AWS Database Migration Service
- Utilisation d'Amazon Kinesis Data Streams comme cible pour AWS Database Migration Service
- Utiliser Apache Kafka comme cible pour AWS Database Migration Service
- Utilisation d'un cluster Amazon OpenSearch Service en tant que cible pour AWS Database Migration Service
- Utilisation d'Amazon DocumentDB en tant que cible pour AWS Database Migration Service
- Utilisation d'Amazon Neptune en tant que cible pour AWS Database Migration Service
- Utilisation de Redis en tant que cible pour AWS Database Migration Service
- Utilisation de Babelfish en tant que cible pour AWS Database Migration Service

- Utilisation d'Amazon Timestream comme cible pour AWS Database Migration Service
- Utilisation d'Amazon RDS pour Db2 et IBM Db2 LUW en tant que cible pour AWS DMS

Utilisation d'une base de données Oracle comme cible pour AWS Database Migration Service

Vous pouvez migrer les données vers les cibles de base de données Oracle à l'aide de AWS DMS, soit depuis une autre base de données Oracle, soit depuis l'une des autres bases de données prises en charge. Vous pouvez utiliser SSL (Secure Sockets Layer) pour chiffrer les connexions entre votre point de terminaison Oracle et l'instance de réplication. Pour plus d'informations sur l'utilisation de SSL avec un point de terminaison Oracle, consultez Utilisation du protocole SSL avec AWS Database Migration Service. AWS DMS prend également en charge l'utilisation de Transparent Data Encryption (TDE) (chiffrement transparent des données) d'Oracle pour chiffrer les données au repos dans la base de données cible, car Oracle TDE n'a pas besoin d'une clé de chiffrement ou d'un mot de passe pour écrire dans la base de données.

Pour en savoir plus sur les versions d'Oracle qui sont prises en charge par AWS DMS en tant que cible, consultez Objectifs pour AWS DMS.

Quand vous utilisez Oracle comme cible, nous supposons que les données doivent être migrées vers le schéma ou l'utilisateur qui est utilisé pour la connexion cible. Si vous voulez migrer des données vers un autre schéma, utilisez une transformation de schéma pour cela. Par exemple, supposons que votre point de terminaison cible se connecte à l'utilisateur `RDSMASTER` et que vous souhaitez migrer de l'utilisateur `PERFDATA1` vers `PERFDATA2`. Dans ce cas, créez une transformation comme suit.

```
{
   "rule-type": "transformation",
   "rule-id": "2",
   "rule-name": "2",
   "rule-action": "rename",
   "rule-target": "schema",
   "object-locator": {
   "schema-name": "PERFDATA1"
},
"value": "PERFDATA2"
}
```

Lorsque vous utilisez Oracle en tant que cible, AWS DMS migre toutes les tables et tous les index vers les espaces de table et d'index par défaut de la cible. Si vous souhaitez migrer des tables et des index vers des espaces de table et d'index différents, utilisez une transformation d'espace de table. Par exemple, supposons que vous avez dans le schéma `INVENTORY` un ensemble de tables affectées à certains espaces de table dans la source Oracle. Pour la migration, vous souhaitez affecter toutes ces tables à un espace de tables unique `INVENTORYSPACE` dans la cible. Dans ce cas, créez une transformation comme suit.

```
{
   "rule-type": "transformation",
   "rule-id": "3",
   "rule-name": "3",
   "rule-action": "rename",
   "rule-target": "table-tablespace",
   "object-locator": {
      "schema-name": "INVENTORY",
      "table-name": "%",
      "table-tablespace-name": "%"
   },
   "value": "INVENTORYSPACE"
}
```

Pour plus d'informations sur les transformations, consultez Spécification des règles de sélection de table et de transformation à l'aide de JSON.

Si Oracle est à la fois source et cible, vous pouvez conserver les affectations d'espace de table de table ou d'index existantes en définissant l'attribut de connexion supplémentaire source Oracle, `enableHomogenousTablespace=true`. Pour de plus amples informations, veuillez consulter Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS.

Pour plus de détails sur l'utilisation des bases de données Oracle comme cible pour AWS DMS, consultez les sections suivantes :

Rubriques

- Limitations sur Oracle comme cible pour AWS Database Migration Service
- Privilèges de compte d'utilisateur requis pour utiliser Oracle comme cible
- Configuration d'une base de données Oracle comme cible pour AWS Database Migration Service
- Paramètres de point de terminaison lors de l'utilisation d'Oracle en tant que cible pour AWS DMS
- Types de données cibles pour Oracle

Limitations sur Oracle comme cible pour AWS Database Migration Service

Les limitations lors de l'utilisation d'Oracle comme cible pour la migration de données sont les suivantes :

- AWS DMS ne crée pas de schéma sur la base de données Oracle cible. Vous devez créer les schémas que vous voulez sur la base de données Oracle cible. Le nom du schéma doit déjà exister pour la cible Oracle. Les tables du schéma source sont importés dans l'utilisateur ou le schéma, que AWS DMS utilise pour se connecter à l'instance cible. Vous pouvez créer plusieurs tâches de réplication si vous avez plusieurs schémas à migrer. Vous pouvez également migrer des données vers différents schémas sur une cible. Pour ce faire, vous devez utiliser des règles de transformation de schéma sur les mappages de table AWS DMS.
- AWS DMS ne prend pas en charge l'option `Use direct path full load` pour les tables avec INDEXTYPE CONTEXT. Pour contourner ce problème, vous pouvez utiliser le chargement de la grappe.
- Avec l'option d'application optimisée par lots, le chargement dans la table de modifications nettes utilise un chemin d'accès direct, qui ne prend pas en charge le type XML. Pour contourner ce problème, vous pouvez utiliser le mode d'application transactionnel.
- Les chaînes vides migrées à partir de bases de données sources peuvent être traitées différemment par la cible Oracle (converties en chaînes d'un espace, par exemple). Cela peut entraîner le signalement d'une non-concordance par la validation AWS DMS.
- Vous pouvez exprimer le nombre total de colonnes par table prises en charge dans le mode d'application par lots optimisée, à l'aide de la formule suivante :

  ```
  2 * columns_in_original_table + columns_in_primary_key <= 999
  ```

  Par exemple, si la table d'origine comporte 25 colonnes et que sa clé primaire est composée de 5 colonnes, le nombre total de colonnes est de 55. Si une table dépasse le nombre de colonnes pris en charge, toutes les modifications sont appliquées une par une.
- AWS DMS ne prend pas en charge la base de données autonome sur Oracle Cloud Infrastructure (OCI).

Privilèges de compte d'utilisateur requis pour utiliser Oracle comme cible

Pour utiliser une cible Oracle dans une tâche AWS Database Migration Service, accordez les privilèges suivants dans la base de données Oracle. Vous accordez ces privilèges au compte utilisateur spécifié dans les définitions de la base de données Oracle pour AWS DMS.

- SELECT ANY TRANSACTION
- SELECT on V$NLS_PARAMETERS
- SELECT on V$TIMEZONE_NAMES
- SELECT on ALL_INDEXES
- SELECT on ALL_OBJECTS
- SELECT on DBA_OBJECTS
- SELECT on ALL_TABLES
- SELECT on ALL_USERS
- SELECT on ALL_CATALOG
- SELECT on ALL_CONSTRAINTS
- SELECT on ALL_CONS_COLUMNS
- SELECT on ALL_TAB_COLS
- SELECT on ALL_IND_COLUMNS
- DROP ANY TABLE
- SELECT ANY TABLE
- INSERT ANY TABLE
- UPDATE ANY TABLE
- CREATE ANY VIEW
- DROP ANY VIEW
- CREATE ANY PROCEDURE
- ALTER ANY PROCEDURE
- DROP ANY PROCEDURE
- CREATE ANY SEQUENCE
- ALTER ANY SEQUENCE
- DROP ANY SEQUENCE
- DELETE ANY TABLE

Pour les exigences suivantes, accordez ces privilèges supplémentaires :

- Pour utiliser une liste de tables spécifique, accordez l'autorisation SELECT sur n'importe quelle table répliquée et ALTER sur n'importe quelle table répliquée.
- Pour permettre à un utilisateur de créer une table dans un espace de table par défaut, accordez-lui le privilège GRANT UNLIMITED TABLESPACE.
- Pour l'ouverture de session, accordez le privilège CREATE SESSION.
- Si vous utilisez un chemin direct (qui est le chemin par défaut pour le chargement complet), `GRANT LOCK ANY TABLE to dms_user;`.
- Si le schéma est différent lors de l'utilisation du mode de préparation de table « DROP and CREATE », `GRANT CREATE ANY INDEX to dms_user;`.
- Pour certains scénarios de chargement complet, vous pouvez choisir l'option « SUPPRIMER et CRÉER une table » ou « TRONQUER avant le chargement » où un schéma de table cible est différent de celui de l'utilisateur DMS. Dans ce cas, accordez SUPPRIMER N'IMPORTE QUELLE TABLE.
- Pour stocker les modifications dans des tables de modification ou une table d'audit où le schéma de table cible est différent de celui de l'utilisateur DMS, accordez CRÉER N'IMPORTE QUELLE TABLE et CRÉER N'IMPORTE QUEL INDEX.

Privilèges de lecture requis pour AWS Database Migration Service sur la base de données cible

Le compte d'utilisateur AWS DMS doit disposer d'autorisations de lecture pour les tables d'administrateur de base de données suivantes :

- SELECT on DBA_USERS
- SELECT on DBA_TAB_PRIVS
- SELECT on DBA_OBJECTS
- SELECT on DBA_SYNONYMS
- SELECT on DBA_SEQUENCES
- SELECT on DBA_TYPES
- SELECT on DBA_INDEXES
- SELECT on DBA_TABLES
- SELECT on DBA_TRIGGERS
- SELECT on SYS.DBA_REGISTRY

Si certains des privilèges requis ne peuvent pas être accordés à V$ xxx, accordez-les à V_$ xxx.

Configuration d'une base de données Oracle comme cible pour AWS Database Migration Service

Avant d'utiliser une base de données Oracle comme cible de la migration des données, vous devez fournir un compte d'utilisateur Oracle à AWS DMS. Le compte d'utilisateur doit disposer de privilèges de lecture/écriture sur la base de données Oracle, comme indiqué au chapitre Privilèges de compte d'utilisateur requis pour utiliser Oracle comme cible.

Paramètres de point de terminaison lors de l'utilisation d'Oracle en tant que cible pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données cible Oracle comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison cible à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--oracle-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec Oracle en tant que cible sont indiqués dans le tableau suivant.

Name (Nom)	Description
`EscapeCharacter`	Définissez cet attribut sur un caractère d'échappement. Ce caractère d'échappement vous permet de faire en sorte qu'un caractère générique unique se comporte comme un caractère normal dans les expressions de mappage de table. Pour de plus amples informati ons, veuillez consulter Caractères génériques dans le mappage de table. Valeur par défaut : Null Valeurs valides : tout caractère autre qu'un caractère générique Exemple : `--oracle-settings '{"Escape Character": "#"}'`

Name (Nom)	Description
`UseDirectPathFullLoad`	Lorsqu'elle est définie sur Y, AWS DMS utilise une charge complète de chemin direct. Spécifiez cette valeur pour activer le protocole de chemin direct dans Oracle Call Interface (OCI). Ce protocole OCI permet le chargement en bloc des tables cibles Oracle pendant un chargement complet. Valeur par défaut : `true` Valeurs valides : `true/false` Exemple : `--oracle-settings '{"UseDirectPathFullLoad": false}'`
`DirectPathParallelLoad`	Lorsqu'il est défini sur `true`, cet attribut spécifie une charge parallèle lorsque la valeur `UseDirectPathFullLoad` est définie sur Y. Cet attribut s'applique également uniquement lorsque vous utilisez la fonction de charge parallèle AWS DMS. Pour de plus amples informations, veuillez consulter l'opération `parallel-load` dans Règles des paramètres de table et de collection et opérations. Une limitation de la spécification de ce paramètre de charge parallèle est que la table cible ne peut pas avoir de contraintes ou d'index. Pour de plus amples informations sur cette limitation, veuillez consulter Activation des contraintes après une charge de chemin directe parallèle. Si des contraintes ou des index sont activés, la définition de cet attribut sur `true` n'a aucun effet. Valeur par défaut : `false` Valeurs valides : `true/false` Exemple : `--oracle-settings '{"DirectPathParallelLoad": true}'`

Name (Nom)	Description
`DirectPathNoLog`	Lorsqu'il est défini sur `true`, cet attribut permet d'augmenter le taux de validation sur la base de données cible Oracle en écrivant directement dans les tables et en n'écrivant pas de trace dans les journaux de base de données. Pour de plus amples informations, veuillez consulter Direct-Load INSERT. Cet attribut s'applique également uniquement lorsque vous définissez `UseDirectPathFullLoad` sur `Y`. Valeur par défaut : `false` Valeurs valides : `true/false` Exemple : `--oracle-settings '{"DirectPathNoLog": true}'`
`CharLengthSemantics`	Spécifie si la longueur d'une colonne de caractères est exprimée en octets ou en caractères. Pour indiquer que la longueur de la colonne de caractères est en caractères, définissez cet attribut sur `CHAR`. Sinon, la longueur de la colonne de caractères est exprimée en octets. Valeur par défaut : Non définie sur `CHAR` Valeurs valides : `CHAR` Exemple : `--oracle-settings '{"CharLengthSemantics": "CHAR"}'`

Name (Nom)	Description
`AlwaysReplaceEmptyString`	AWS DMS ajoute un espace supplémentaire pour répliquer une chaîne vide lors de la migration vers une cible Oracle. En général, Oracle n'a pas de notation pour une chaîne vide. Lorsque vous insérez une chaîne vide au format varchar2, vous chargez des chaînes vides au format NULL. Si vous souhaitez insérer les données au format NULL sur Oracle, définissez cet attribut sur FALSE. Valeur par défaut : `true` Valeurs valides : `true/false` Exemple : `--oracle-settings '{"Always ReplaceEmptyString": false}'`

Types de données cibles pour Oracle

Une base de données Oracle cible utilisée avec AWS DMS prend en charge la plupart des types de données Oracle. Le tableau suivant présente les types de données cibles Oracle qui sont pris en charge lorsque vous utilisez AWS DMS et le mappage par défaut à partir des types de données AWS DMS. Pour en savoir plus sur la façon d'afficher le type de données qui est mappé à partir de la source, consultez la section correspondant à la source que vous utilisez.

Type de données AWS DMS	Type de données Oracle
BOOLEAN	NUMBER (1)
BYTES	RAW (length)
DATE	DATETIME
TIME	TIMESTAMP (0)
DATETIME	TIMESTAMP (scale)

Type de données AWS DMS	Type de données Oracle
INT1	NUMBER (3)
INT2	NUMBER (5)
INT4	NUMBER (10)
INT8	NUMBER (19)
NUMERIC	NUMBER (p,s)
REAL4	FLOAT
REAL8	FLOAT
CHAÎNE	Avec indication date : DATE Avec indication time : TIMESTAMP Avec indication timestamp : TIMESTAMP Avec indication timestamp_with_timezone : TIMESTAMP WITH TIMEZONE Avec indication timestamp_with_local_timezone : TIMESTAMP WITH LOCAL TIMEZONE Avec indication interval_year_to_month : INTERVAL YEAR TO MONTH Avec indication interval_day_to_second : INTERVAL DAY TO SECOND Si la longueur > 4 000 : CLOB Dans tous les autres cas : VARCHAR2 (longueur)
UINT1	NUMBER (3)
UINT2	NUMBER (5)

Type de données AWS DMS	Type de données Oracle
UINT4	NUMBER (10)
UINT8	NUMBER (19)
WSTRING	Si la longueur > 2 000 : NCLOB Dans tous les autres cas : NVARCHAR2 (longueur)
BLOB	BLOB Pour pouvoir utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation des types de données BLOB pour une tâche spécifique. Les types de données BLOB sont pris en charge uniquement dans les tables qui contiennent une clé primaire
CLOB	CLOB Pour pouvoir utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation des types de données CLOB pour une tâche spécifique. Au cours de la capture des données modifiées (CDC), les types de données CLOB sont pris en charge uniquement dans les tables qui comprennent une clé primaire. CHAÎNE Un type de données Oracle VARCHAR2 sur la source dont la taille déclarée est supérieure à 4 000 octets est mappé via l'objet CLOB AWS DMS à une STRING sur la cible Oracle.

Type de données AWS DMS	Type de données Oracle
NCLOB	NCLOB Pour pouvoir utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation des types de données NCLOB pour une tâche spécifique. Au cours de la capture des données modifiées (CDC), les types de données NCLOB sont pris en charge uniquement dans les tables qui contiennent une clé primaire. WSTRING Un type de données Oracle VARCHAR2 sur la source dont la taille déclarée est supérieure à 4 000 octets est mappé via l'objet NCLOB AWS DMS à une WSTRING sur la cible Oracle.
XMLTYPE	Le type de données cible XMLTYPE est pertinent uniquement dans les tâches de réplication d'Oracle à Oracle. Lorsque la base de données source est Oracle, les types de donnée source sont répliqués en l'état dans la cible Oracle. Par exemple, un type de données XMLTYPE dans la source est créé en tant que type de données XMLTYPE dans la cible.

Utilisation d'une base de données Microsoft SQL Server comme cible pour AWS Database Migration Service

Vous pouvez migrer les données vers les bases de données Microsoft SQL Server à l'aide de AWS DMS. Avec une base de données SQL Server comme cible, vous pouvez migrer les données à partir d'une autre base de données SQL Server, ou d'une des autres bases de données prises en charge.

Pour en savoir plus sur les versions de SQL Server qui sont prises en charge par AWS DMS en tant que cible, consultez Objectifs pour AWS DMS.

AWS DMS prend en charge les éditions sur site ainsi que les versions Enterprise, Standard, Workgroup, Developer et Web d'Amazon RDS.

Pour plus de détails sur l'utilisation de AWS DMS et des bases de données cibles SQL Server, consultez les ressources suivantes.

Rubriques

- Limitations sur l'utilisation de SQL Server comme cible pour AWS Database Migration Service
- Exigences de sécurité lors de l'utilisation de SQL Server comme cible pour AWS Database Migration Service
- Paramètres de point de terminaison lors de l'utilisation de SQL Server en tant que cible pour AWS DMS
- Types de données cibles pour Microsoft SQL Server

Limitations sur l'utilisation de SQL Server comme cible pour AWS Database Migration Service

Les limitations suivantes s'appliquent lors de l'utilisation d'une base de données SQL Server comme cible pour AWS DMS :

- Lorsque vous créez manuellement une table SQL Server cible avec une colonne calculée, la réplication de chargement complet n'est pas prise en charge lors de l'utilisation de l'utilitaire de copie groupée BCP. Pour utiliser la réplication à chargement complet, désactivez le chargement BCP en définissant l'attribut de connexion supplémentaire (ECA) `'useBCPFullLoad=false'` sur le point de terminaison. Pour en savoir plus sur la définition d'ECA sur les points de terminaison, consultez Création de points de terminaison source et cible. Pour en savoir plus sur l'utilisation avec BCP, consultez la documentation Microsoft SQL Server.
- Lors de la réplication de tables avec des types de données spatiales SQL Server (GEOMETRY et GEOGRAPHY), AWS DMS remplace tous les identifiants de référence spatiale (SRID) que vous pouvez avoir insérés par le SRID par défaut. Le SRID par défaut est 0 pour GEOMETRY et 4326 pour GEOGRAPHY.
- Les tables temporelles ne sont pas prises en charge. La migration de tables temporelles peut fonctionner avec une tâche de réplication seule en mode d'application transactionnelle si ces tables sont créées manuellement sur la cible.
- Actuellement, les types de données `boolean` dans une source PostgreSQL sont migrés vers une cible SQLServer en tant que type de données `bit` avec des valeurs incohérentes.

  Pour contourner ce problème, procédez comme suit :

  - Pré-créez la table au préalable avec un type de données `VARCHAR(1)` pour la colonne (ou laissez AWS DMS créer la table). Ensuite, faites en sorte que le traitement en aval traite un « F » comme Faux et un « T » comme Vrai.
  - Pour éviter d'avoir à modifier le traitement en aval, ajoutez une règle de transformation à la tâche pour remplacer les valeurs « F » par « 0 » et les valeurs « T » par 1, puis stockez-les en tant que type de données binaire SQL Server.
- AWS DMS ne prend pas en charge le traitement des modifications pour définir la possibilité de valeur NULL des colonnes (à l'aide de la clause `ALTER COLUMN [SET|DROP] NOT NULL` avec les instructions `ALTER TABLE`).
- L'authentification Windows n'est pas prise en charge.

Exigences de sécurité lors de l'utilisation de SQL Server comme cible pour AWS Database Migration Service

La section suivante décrit les exigences de sécurité pour l'utilisation de AWS DMS avec une cible de Microsoft SQL Server :

- Le compte utilisateur AWS DMS doit avoir au moins le rôle utilisateur `db_owner` sur la base de données SQL Server à laquelle vous êtes connecté.
- Un administrateur système SQL Server doit fournir cette autorisation à tous les comptes utilisateur AWS DMS.

Paramètres de point de terminaison lors de l'utilisation de SQL Server en tant que cible pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données cible SQL Server comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison cible à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--microsoft-sql-server-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec SQL Server en tant que cible sont indiqués dans le tableau suivant.

Nom	Description

`ControlTablesFileGroup`	Spécifier un groupe de fichiers pour les tables internes de AWS DMS. Lorsque la tâche de réplication commence, toutes les tables de contrôle AWS DMS internes (awsdms_ apply_exception, awsdms_apply, awsdms_changes) sont créées dans le groupe de fichiers spécifié.
	
	Valeur par défaut : s/o
	
	Valeurs valides : string
	
	Exemple : `--microsoft-sql-server-settings '{"ControlTablesFileGroup": "filegroup1"}'`
	
	Voici un exemple d'une commande pour créer un groupe de fichiers.
	
	`ALTER DATABASE replicate ADD FILEGROUP Test1FG1;`
	`GO ALTER DATABASE replicate`
	`ADD FILE (`
	`NAME = test1dat5,`
	`FILENAME = 'C:\temp\DATA\t1dat5.ndf',`
	
	`SIZE = 5MB,`
	`MAXSIZE = 100MB,`
	`FILEGROWTH = 5MB`
	`)`
	`TO FILEGROUP Test1FG1;`
	`GO`

`ExecuteTimeout`	Utilisez cet attribut de connexion supplémentaire (ECA) pour définir le délai d'expiration de l'instruction client pour l'instance SQL Server, en secondes. La valeur par défaut est de 60 secondes. Exemple : `'{"ExecuteTimeout": 100}'`

Nom	Description
`UseBCPFullLoad`	Utilisez cet attribut pour transférer les données pour les opérations de chargement complet à l'aide de BCP. Lorsque la table cible contient une colonne d'identité qui n'existe pas dans la table source, vous devez désactiver l'option Utiliser BCP pour charger la table. Valeur par défaut : true Valeurs valides : true/false Exemple : `--microsoft-sql-server-settings '{"UseBCPFullLoad": false}'`

Types de données cibles pour Microsoft SQL Server

Le tableau suivant présente les types de données cibles Microsoft SQL Server qui sont pris en charge lorsque vous utilisez AWS DMS et le mapping par défaut des types de données AWS DMS. Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Type de données AWS DMS	Type de données SQL Server
BOOLEAN	TINYINT
BYTES	VARBINARY(Length)
DATE	Pour SQL Server 2008 et versions ultérieures, utilisez DATE. Pour les versions antérieures, si l'échelle est 3 ou inférieure utilisez DATETIME. Dans tous les autres cas, utilisez VARCHAR (37).
TIME	Pour SQL Server 2008 et versions ultérieures, utilisez DATETIME2 (%d).

Type de données AWS DMS	Type de données SQL Server
	Pour les versions antérieures, si l'échelle est 3 ou inférieure utilisez DATETIME. Dans tous les autres cas, utilisez VARCHAR (37).
DATETIME	Pour SQL Server 2008 et versions ultérieures, utilisez DATETIME2 (scale). Pour les versions antérieures, si l'échelle est 3 ou inférieure utilisez DATETIME. Dans tous les autres cas, utilisez VARCHAR (37).
INT1	SMALLINT
INT2	SMALLINT
INT4	INT
INT8	BIGINT
NUMERIC	NUMERIC (p,s)
REAL4	REAL
REAL8	FLOAT
CHAÎNE	Si la colonne est une colonne de date ou d'heure, effectuez les opérations suivantes : • Pour SQL Server 2008 et versions ultérieures, utilisez DATETIME2. • Pour les versions antérieures, si l'échelle est 3 ou inférieure utilisez DATETIME. Dans tous les autres cas, utilisez VARCHAR (37). Si la colonne n'est pas une colonne de date ou d'heure, utilisez VARCHAR (length).
UINT1	TINYINT

Type de données AWS DMS	Type de données SQL Server
UINT2	SMALLINT
UINT4	INT
UINT8	BIGINT
WSTRING	NVARCHAR (length)
BLOB	VARBINARY(max) IMAGE Pour pouvoir utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation des BLOB pour une tâche spécifique. AWS DMS prend en charge les types de données BLOB uniquement dans les tables qui contiennent une clé primaire.
CLOB	VARCHAR(max) Pour pouvoir utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation des types de données CLOB pour une tâche spécifique. Au cours de la capture des données modifiées (CDC), AWS DMS prend en charge les types de données CLOB uniquement dans les tables qui comprennent une clé primaire.
NCLOB	NVARCHAR(max) Pour pouvoir utiliser ce type de données avec AWS DMS, vous devez activer l'utilisation des types de données NCLOB pour une tâche spécifique. Au cours de la capture des données modifiées (CDC), AWS DMS prend en charge les types de données NCLOB uniquement dans les tables qui contiennent une clé primaire.

Utilisation d'une base de données PostgreSQL comme cible pour AWS Database Migration Service

Vous pouvez migrer des données vers des bases de données PostgreSQL à AWS DMS l'aide d'une autre base de données PostgreSQL ou de l'une des autres bases de données prises en charge.

Pour plus d'informations sur les versions de PostgreSQL compatibles en tant AWS DMS que cible, consultez. Objectifs pour AWS DMS

AWS DMS version à utiliser
Utilisez n'importe quelle AWS DMS version disponible.
Utilisez AWS DMS la version 3.4.3 et supérieure.
Utilisez AWS DMS la version 3.4.7 ou supérieure.
Utilisez AWS DMS la version 3.5.1 ou supérieure.

Note

- Amazon Aurora sans serveur est disponible en tant que CIBLE pour Amazon Aurora avec compatibilité PostgreSQL. Pour plus d'informations sur Amazon Aurora Serverless, consultez la section Utilisation d'Amazon Aurora Serverless v2 dans le guide de l'utilisateur Amazon Aurora.
- Les clusters de bases de données Aurora sans serveur sont uniquement accessibles à partir d'un réseau Amazon VPC et ne peuvent pas utiliser d'adresse IP publique. Par conséquent, si vous avez l'intention de disposer d'une instance de réplication dans une région différente de celle d'Aurora PostgreSQL sans serveur, vous devez configurer l'appairage de VPC. Sinon, vérifiez la disponibilité des régions Aurora PostgreSQL sans

serveur et décidez d'utiliser l'une de ces régions à la fois pour Aurora PostgreSQL sans serveur et pour votre instance de réplication.

- La fonctionnalité Babelfish est intégrée à Amazon Aurora et n'entraîne aucun coût supplémentaire. Pour plus d'informations, consultez Utilisation de Babelfish for Aurora PostgreSQL en tant que cible pour AWS Database Migration Service.

AWS DMS adopte une table-by-table approche lors de la migration des données de la source vers la cible pendant la phase de chargement complet. L'ordre des tables pendant la phase de chargement complet ne peut pas être garanti. Les tables ne sont pas synchronisées pendant la phase de chargement complet et pendant que les transactions mises en cache pour les tables individuelles sont appliquées. En conséquence, les contraintes actives d'intégrité référentielle peuvent entraîner l'échec d'une tâche pendant la phase de chargement complet.

Dans PostgreSQL, les clés étrangères (contraintes d'intégrité référentielle) sont mises en œuvre à l'aide de déclencheurs. Pendant la phase de chargement complet, AWS DMS charge chaque table une par une. Nous vous recommandons vivement de désactiver les contraintes de clé étrangère pendant un chargement complet, à l'aide de l'une des méthodes suivantes :

- Désactivez temporairement tous les déclencheurs depuis l'instance, et terminez le chargement complet.
- Utilisez le paramètre `session_replication_role` dans PostgreSQL.

À un instant donné, un déclencheur peut être dans l'un des états suivants : `origin`, `replica`, `always` ou `disabled`. Lorsque le paramètre `session_replication_role` a la valeur `replica`, seuls les déclencheurs ayant l'état `replica` sont actifs et sont déclenchés lorsqu'ils sont appelés. Sinon, les déclencheurs demeurent inactifs.

PostgreSQL possède un mécanisme failsafe pour empêcher qu'une table soit tronquée, même lorsque `session_replication_role` est défini. Vous pouvez l'utiliser comme alternative à la désactivation des déclencheurs, afin d'aider le chargement complet à s'exécuter intégralement. Pour ce faire, définissez le mode de préparation des tables cible avec la valeur `DO_NOTHING`. Dans le cas contraire, les opérations DROP et TRUNCATE échouent lorsqu'il existe des contraintes de clé étrangère.

Dans Amazon RDS, vous pouvez définir ce paramètre à l'aide d'un groupe de paramètres. Pour une instance PostgreSQL s'exécutant sur Amazon EC2, vous pouvez définir le paramètre directement.

Pour plus d'informations sur l'utilisation d'une base de données PostgreSQL en tant que cible AWS DMS pour, consultez les sections suivantes :

Rubriques

- Limitations relatives à l'utilisation de PostgreSQL comme cible pour AWS Database Migration Service
- Exigences de sécurité lors de l'utilisation d'une base de données PostgreSQL comme cible pour AWS Database Migration Service
- Paramètres du point de terminaison et attributs de connexion supplémentaires (ECA) lors de l'utilisation de PostgreSQL comme cible pour AWS DMS
- Types de données cibles pour PostgreSQL
- Utilisation de Babelfish pour Aurora PostgreSQL comme cible pour AWS Database Migration Service

Limitations relatives à l'utilisation de PostgreSQL comme cible pour AWS Database Migration Service

Les limitations suivantes s'appliquent lors de l'utilisation d'une base de données PostgreSQL comme cible pour AWS DMS :

- Pour les migrations hétérogènes, le type de données JSON est converti en type de données CLOB natif en interne.
- Lors d'une migration d'Oracle vers PostgreSQL, si une colonne d'Oracle contient un caractère NULL (valeur hexadécimale U+0000) AWS DMS , le caractère NULL est converti en espace (valeur hexadécimale U+0020). Cela est dû à une limitation PostgreSQL.
- AWS DMS ne prend pas en charge la réplication vers une table avec un index unique créé avec la fonction de coalesce.
- Si vos tables utilisent des séquences, mettez à jour la valeur de `NEXTVAL` pour chaque séquence dans la base de données cible après avoir arrêté la réplication depuis la base de données source. AWS DMS copie les données de votre base de données source, mais ne migre pas les séquences vers la cible pendant la réplication en cours.

Exigences de sécurité lors de l'utilisation d'une base de données PostgreSQL comme cible pour AWS Database Migration Service

À des fins de sécurité, le compte d'utilisateur utilisé pour la migration des données doit être un utilisateur enregistré dans une base de données PostgreSQL que vous utilisez comme cible.

Votre point de terminaison cible PostgreSQL nécessite des autorisations utilisateur minimales pour exécuter AWS DMS une migration, consultez les exemples suivants.

```
    CREATE USER newuser WITH PASSWORD 'your-password';
    ALTER SCHEMA schema_name OWNER TO newuser;
```

Ou

```
    GRANT USAGE ON SCHEMA schema_name TO myuser;
    GRANT CONNECT ON DATABASE postgres to myuser;
    GRANT CREATE ON DATABASE postgres TO myuser;
    GRANT CREATE ON SCHEMA schema_name TO myuser;
    GRANT UPDATE, INSERT, SELECT, DELETE, TRUNCATE ON ALL TABLES IN SCHEMA schema_name
TO myuser;
    GRANT TRUNCATE ON schema_name."BasicFeed" TO myuser;
```

Paramètres du point de terminaison et attributs de connexion supplémentaires (ECA) lors de l'utilisation de PostgreSQL comme cible pour AWS DMS

Vous pouvez utiliser les paramètres des points de terminaison et les attributs de connexion supplémentaires (ECA) pour configurer votre base de données cible PostgreSQL.

Vous spécifiez les paramètres lorsque vous créez le point de terminaison cible à l'aide de la AWS DMS console ou à l'aide de la `create-endpoint` commande dans le AWS CLI, avec la syntaxe `--postgre-sql-settings '{"EndpointSetting": "value", ...}'` JSON.

Vous spécifiez les ECA à l'aide du `ExtraConnectionAttributes` paramètre de votre point de terminaison.

Les paramètres de point de terminaison que vous pouvez utiliser avec PostgreSQL en tant que cible sont indiqués dans le tableau suivant.

Name (Nom)	Description
`MaxFileSize`	Spécifie la taille maximale (en Ko) de tout fichier .csv utilisé pour transférer des données vers PostgreSQL. Valeur par défaut : 32 768 Ko (32 Mo) Valeurs valides : 1 à 1 048 576 Ko (jusqu'à 1,1 Go) Exemple : `--postgre-sql-settings '{"MaxFileSize": 512}'`
`ExecuteTimeout`	Définit le délai d'attente de déclaration client pour l'instance PostgreSQL, en secondes. La valeur par défaut est de 60 secondes. Exemple : `--postgre-sql-settings '{"ExecuteTimeout": 100}'`
`AfterConnectScript= SET session_replication_role = replica`	Cet attribut permet de AWS DMS contourner les clés étrangères et les déclencheurs utilisateur afin de réduire le temps nécessaire au chargement groupé des données.
`MapUnboundedNumericAsString`	Ce paramètre traite les colonnes contenant des types de données NUMERIC illimités comme STRING afin de réussir la migration sans perte de précision de la valeur numérique. Utilisez ce paramètre uniquement pour la réplication d'une source PostgreSQL vers une cible PostgreSQL ou pour des bases de données compatibles avec PostgreSQL. Valeur par défaut : false Valeurs valides : false/true Exemple : `--postgre-sql-settings '{"MapUnboundedNumericAsString": "true"}` L'utilisation de ce paramètre peut entraîner une certaine dégradation des performances de réplication en raison

Name (Nom)	Description
	de la transformation de la valeur numérique en chaîne, puis à nouveau en valeur numérique. Ce paramètre est compatible avec DMS versions 3.4.4 et ultérieures. **Note** Utilisez `MapUnboundedNumericAsString` uniquement dans les points de terminaison source et cible PostgreSQL ensemble. L'utilisation de `MapUnboundedNumericAsString` sur les points de terminaison PostgreSQL source limite la précision à 28 pendant la CDC. L'utilisation de `MapUnboundedNumericAsString` sur les points de terminaison cibles migre les données avec une précision de 28 et une échelle de 6. N'utilisez pas `MapUnboundedNumericAsString` avec des cibles autres que PostgreSQL.
`loadUsingCSV`	Utilisez cet attribut de connexion supplémentaire (ECA) pour transférer des données pour les opérations de chargement complet à l'aide de la commande \ COPY. Valeur par défaut : `true` Valeurs valides : true/false Exemple d'ECA : `loadUsingCSV=true;` Remarque : le fait de définir cet ECA sur false peut entraîner une certaine dégradation des performances de réplication en raison de l'exécution directe des INSERTS.

Name (Nom)	Description
`DatabaseMode`	Utilisez cet attribut pour modifier le comportement par défaut de la gestion par la réplication des points de terminaison compatibles avec Postgresql qui nécessitent une configuration supplémentaire, tels que les points de terminaison Babelfish. Valeur par défaut : `DEFAULT` Valeurs valides : `DEFAULT`, `BABELFISH` Exemple : `DatabaseMode=default;`
`BabelfishDatabaseName`	Utilisez cet attribut pour spécifier le nom de la base de données T-SQL Babelfish cible vers laquelle la migration est effectuée. Obligatoire si `DatabaseMode` est défini sur `Babelfish` . Il ne s'agit pas de la base de données `babelfish_db` réservée. Exemple : `BabelfishDatabaseName=TargetDb;`

Types de données cibles pour PostgreSQL

Le point de terminaison de base de données PostgreSQL pour AWS DMS prend en charge la plupart des types de données de base de données PostgreSQL. Le tableau suivant indique les types de données cibles de base de données PostgreSQL pris en charge lors de l' AWS DMS utilisation et le mappage AWS DMS par défaut à partir des types de données.

Pour plus d'informations sur AWS DMS les types de données, consultezTypes de données pour AWS Database Migration Service.

AWS DMS type de données	Type de données PostgreSQL
BOOLEAN	BOOLEAN
BLOB	BYTEA

AWS DMS type de données	Type de données PostgreSQL
BYTES	BYTEA
DATE	DATE
TIME	TIME
DATETIME	Si l'échelle s'étend de 0 à 6, utilisez TIMESTAMP. Si l'échelle s'étend de 7 à 9, utilisez VARCHAR (37).
INT1	SMALLINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	DECIMAL (P,S)
REAL4	FLOAT4
REAL8	FLOAT8
CHAÎNE	Si la longueur est comprise entre 1 et 21 845, utilisez VARCHAR (longueur en octets). Si la longueur est comprise entre 21 846 et 2 147 483 647, utilisez VARCHAR (65535).
UINT1	SMALLINT
UINT2	INTEGER
UINT4	BIGINT
UINT8	BIGINT

AWS DMS type de données	Type de données PostgreSQL
WSTRING	Si la longueur est comprise entre 1 et 21 845, utilisez VARCHAR (longueur en octets). Si la longueur est comprise entre 21 846 et 2 147 483 647, utilisez VARCHAR (65535).
NCLOB	TEXT
CLOB	TEXT

> **Note**
>
> Lors de la réplication à partir d'une source PostgreSQL AWS DMS , crée la table cible avec les mêmes types de données pour toutes les colonnes, à l'exception des colonnes avec des types de données définis par l'utilisateur. Dans ces cas, le type de données est créé en tant que « character varying » dans la cible.

Utilisation de Babelfish pour Aurora PostgreSQL comme cible pour AWS Database Migration Service

Vous pouvez migrer les tables sources SQL Server vers une cible Babelfish for Amazon Aurora PostgreSQL à l'aide d' AWS Database Migration Service. Avec Babelfish, Aurora PostgreSQL comprend T-SQL, le dialecte SQL propriétaire de Microsoft SQL Server, et prend en charge le même protocole de communication. Ainsi, les applications écrites pour SQL Server peuvent désormais fonctionner avec Aurora avec moins de modifications de code. La fonctionnalité Babelfish est intégrée à Amazon Aurora et n'entraîne aucun coût supplémentaire. Vous pouvez activer Babelfish sur votre cluster Amazon Aurora à partir de la console Amazon RDS.

Lorsque vous créez votre point de terminaison AWS DMS cible à l'aide des commandes de AWS DMS console, d'API ou de CLI, spécifiez le moteur cible comme Amazon Aurora PostgreSQL et nommez la base de données babelfish_db. Dans la section Paramètres de point de terminaison, ajoutez les paramètres pour définir `DatabaseMode` sur `Babelfish` et `BabelfishDatabaseName` sur le nom de la base de données Babelfish T-SQL cible.

Ajout de règles de transformation à votre tâche de migration

Lorsque vous définissez une tâche de migration pour une cible Babelfish, vous devez inclure des règles de transformation garantissant que DMS utilise les tables Babelfish T-SQL pré-créées dans la base de données cible.

Commencez par ajouter une règle de transformation à votre tâche de migration qui convertit tous les noms de table en minuscules. Babelfish stocke en minuscules dans le catalogue `pg_class` PostgreSQL les noms des tables que vous créez à l'aide de T-SQL. Toutefois, lorsque vous avez des tables SQL Server avec des noms en casse mixte, DMS crée les tables en utilisant les types de données natifs PostgreSQL au lieu des types de données compatibles T-SQL. Pour cette raison, veillez à ajouter une règle de transformation qui convertit tous les noms de table en minuscules. Notez que les noms de colonnes ne doivent pas être convertis en minuscules.

Ensuite, si vous avez utilisé le mode de migration à plusieurs bases de données lorsque vous avez défini votre cluster, ajoutez une règle de transformation qui renomme le schéma SQL Server d'origine. Assurez-vous de renommer le nom du schéma SQL Server de sorte à inclure le nom de la base de données T-SQL. Par exemple, si le nom du schéma SQL Server d'origine est `dbo` et que le nom de la base de données T-SQL est `mydb`, renommez le schéma `mydb_dbo` en utilisant une règle de transformation.

Si vous utilisez le mode à une seule base de données, vous n'avez pas besoin d'une règle de transformation pour renommer les noms de schéma. Les noms de schéma sont one-to-one mappés avec la base de données T-SQL cible dans Babelfish.

L'exemple de règle de transformation suivant convertit tous les noms de table en minuscules et renomme le nom du schéma SQL Server d'origine `dbo` en `mydb_dbo`.

```
{
   "rules": [
   {
      "rule-type": "transformation",
      "rule-id": "566251737",
      "rule-name": "566251737",
      "rule-target": "schema",
      "object-locator": {
         "schema-name": "dbo"
      },
      "rule-action": "rename",
      "value": "mydb_dbo",
```

```
      "old-value": null
   },
   {
      "rule-type": "transformation",
      "rule-id": "566139410",
      "rule-name": "566139410",
      "rule-target": "table",
      "object-locator": {
         "schema-name": "%",
         "table-name": "%"
      },
      "rule-action": "convert-lowercase",
      "value": null,
      "old-value": null
   },
   {
      "rule-type": "selection",
      "rule-id": "566111704",
      "rule-name": "566111704",
      "object-locator": {
         "schema-name": "dbo",
         "table-name": "%"
      },
      "rule-action": "include",
      "filters": []
   }
]
}
```

Limitations de l'utilisation d'un point de terminaison cible PostgreSQL avec des tables Babelfish

Les limitations suivantes s'appliquent lors de l'utilisation d'un point de terminaison cible PostgreSQL avec des tables Babelfish :

- Pour le mode Préparation de table cible, utilisez uniquement les modes Ne rien faire ou Tronquer. N'utilisez pas le mode Supprimer les tables sur la cible. Dans ce mode, DMS crée les tables en tant que tables PostgreSQL que T-SQL peut ne pas reconnaître.
- AWS DMS ne prend pas en charge le type de données sql_variant.
- Babelfish ne prend pas en charge les types de données `HEIRARCHYID`, `GEOMETRY` et `GEOGRAPHY`. Pour migrer ces types de données, vous pouvez ajouter des règles de transformation pour convertir le type de données en `wstring(250)`.

- Babelfish ne prend en charge que la migration des types de données `BINARY`, `VARBINARY` et `IMAGE` avec le type de données `BYTEA`. Pour les versions antérieures d'Aurora PostgreSQL, vous pouvez utiliser DMS pour migrer ces tables vers un point de terminaison cible Babelfish. Il n'est pas nécessaire de spécifier une longueur pour le type de données `BYTEA`, comme illustré dans l'exemple suivant.

  ```
  [Picture] [VARBINARY](max) NULL
  ```

  Remplacez le type de données T-SQL précédent par le type de données `BYTEA` pris en charge par T-SQL.

  ```
  [Picture] BYTEA NULL
  ```

- Pour les versions antérieures d'Aurora PostgreSQL Babelfish, si vous créez une tâche de migration pour une réplication continue de SQL Server vers Babelfish à l'aide du point de terminaison cible PostgreSQL, vous devez affecter le type de données `SERIAL` à toutes les tables utilisant des colonnes `IDENTITY`. À partir d'Aurora PostgreSQL (versions 15.3/14.8 et ultérieures) et de Babelfish (versions 3.2.0 et ultérieures), la colonne d'identité est prise en charge et il n'est plus nécessaire d'affecter le type de données SERIAL. Pour plus d'informations, consultez Syntaxe de SERIAL dans la section Séquences et identité du Manuel de migration de SQL Server vers Aurora PostgreSQL. Ensuite, lorsque vous créez la table dans Babelfish, modifiez la définition de colonne comme suit.

  ```
      [IDCol] [INT] IDENTITY(1,1) NOT NULL PRIMARY KEY
  ```

  Modifiez la ligne précédente comme suit.

  ```
      [IDCol] SERIAL PRIMARY KEY
  ```

  Aurora PostgreSQL compatible avec Babelfish crée une séquence en utilisant la configuration par défaut et ajoute une contrainte `NOT NULL` à la colonne. La séquence récemment créée se comporte comme une séquence normale (incrémentée de 1) et ne possède aucune option `SERIAL` composite.

- Après avoir migré les données avec des tables qui utilisent des colonnes `IDENTITY` ou le type de données `SERIAL`, réinitialisez l'objet de séquence basé sur PostgreSQL en fonction de la valeur maximale de la colonne. Après avoir effectué un chargement complet des tables, utilisez la requête T-SQL suivante pour générer des instructions afin d'amorcer l'objet de séquence associé.

```
DECLARE @schema_prefix NVARCHAR(200) = ''

IF current_setting('babelfishpg_tsql.migration_mode') = 'multi-db'
        SET @schema_prefix = db_name() + '_'

SELECT 'SELECT setval(pg_get_serial_sequence(''' + @schema_prefix +
 schema_name(tables.schema_id) + '.' + tables.name + ''', ''' + columns.name + ''')
               ,(select max(' + columns.name + ') from ' +
 schema_name(tables.schema_id) + '.' + tables.name + '));'
FROM sys.tables tables
JOIN sys.columns columns ON tables.object_id = columns.object_id
WHERE columns.is_identity = 1

UNION ALL

SELECT 'SELECT setval(pg_get_serial_sequence(''' + @schema_prefix + table_schema +
 '.' + table_name + ''',
''' + column_name + '''),(select max(' + column_name + ') from ' + table_schema + '.'
 + table_name + '));'
FROM information_schema.columns
WHERE column_default LIKE 'nextval(%';
```

La requête génère une série d'instructions SELECT que vous exécutez afin de mettre à jour les valeurs maximales d'IDENTITY et de SERIAL.

- Pour les versions de Babelfish antérieures à 3.2, le Mode LOB complet peut entraîner une erreur de table. Dans ce cas, créez une tâche distincte pour les tables qui n'ont pas pu être chargées. Utilisez ensuite le Mode LOB limité pour spécifier la valeur appropriée pour Taille de LOB maximale (Ko). Sinon, vous pouvez définir le paramètre de l'attribut de connexion du point de terminaison SQL Server`ForceFullLob=True`.
- Pour les versions de Babelfish antérieures à 3.2, la validation des données avec des tables Babelfish qui n'utilisent pas de clés primaires basées sur des entiers génère un message indiquant qu'aucune clé unique appropriée n'a été trouvée. À partir d'Aurora PostgreSQL (versions 15.3/14.8 et ultérieures) et de Babelfish (versions 3.2.0 et ultérieures), la validation des données pour les clés primaires non entières est prise en charge.
- En raison des différences de précision dans le nombre de décimales par seconde, DMS signale des échecs de validation des données pour les tables Babelfish qui utilisent les types de données `DATETIME`. Pour éviter ces échecs, vous pouvez ajouter le type de règle de validation suivant pour les types de données `DATETIME`.

```
{
         "rule-type": "validation",
         "rule-id": "3",
         "rule-name": "3",
         "rule-target": "column",
         "object-locator": {
             "schema-name": "dbo",
             "table-name": "%",
             "column-name": "%",
             "data-type": "datetime"
         },
         "rule-action": "override-validation-function",
         "source-function": "case when ${column-name} is NULL then NULL else 0 end",
         "target-function": "case when ${column-name} is NULL then NULL else 0 end"
     }
```

Utilisation d'une base de données compatible MySQL comme cible pour AWS Database Migration Service

Vous pouvez migrer des données vers toute base de données compatible MySQL à l'aide d'AWS DMS, à partir d'un moteur de données source quelconque pris en charge par AWS DMS. Si vous effectuez la migration vers une base de données sur site compatible MySQL, AWS DMS exige que le moteur source réside dans l'écosystème AWS. Le moteur peut se trouver sur un service géré par AWS tel qu'Amazon RDS, Amazon Aurora ou Amazon S3. Sinon, le moteur peut se trouver sur une base de données autogérée sur Amazon EC2.

Vous pouvez utiliser SSL pour chiffrer les connexions entre votre point de terminaison compatible MySQL et l'instance de réplication. Pour plus d'informations sur l'utilisation de SSL avec un point de terminaison compatible MySQL, consultez Utilisation du protocole SSL avec AWS Database Migration Service.

Pour en savoir plus sur les versions de MySQL qui sont prises en charge par AWS DMS en tant que cible, consultez Objectifs pour AWS DMS.

Vous pouvez utiliser les bases de données compatibles MySQL suivantes en tant que cibles pour AWS DMS :

- MySQL Community Edition

- MySQL Standard Edition
- MySQL Enterprise Edition
- MySQL Cluster Carrier Grade Edition
- MariaDB Community Edition
- MariaDB Enterprise Edition
- MariaDB Column Store
- Amazon Aurora MySQL

> **Note**
>
> Quel que soit le moteur de stockage source (MyISAM, MEMORY, etc.), AWS DMS crée une table cible compatible MySQL en tant que table InnoDB par défaut.
> Si vous avez besoin d'une table qui utilise un moteur de stockage autre qu'InnoDB, vous pouvez manuellement créer la table sur la cible compatible avec MySQL et migrer la table à l'aide de l'option Ne rien faire. Pour de plus amples informations, veuillez consulter Paramètres de tâche de chargement complet.

Pour plus de détails sur l'utilisation d'une base de données compatible MySQL comme cible pour AWS DMS, consultez les sections suivantes.

Rubriques

- Utilisation de n'importe quelle base de données compatible MySQL comme cible pour AWS Database Migration Service
- Limitations sur l'utilisation d'une base de données compatible MySQL comme cible pour AWS Database Migration Service
- Paramètres de point de terminaison lors de l'utilisation d'une base de données compatible MySQL en tant que cible pour AWS DMS
- Types de données cibles pour MySQL

Utilisation de n'importe quelle base de données compatible MySQL comme cible pour AWS Database Migration Service

Avant de commencer à utiliser une base de données compatible avec MySQL comme cible pour AWS DMS, assurez-vous que les conditions préalables suivantes sont remplies :

- Indiquez à AWS DMS un compte utilisateur qui dispose des privilèges de lecture/écriture sur la base de données compatible avec MySQL. Pour créer les privilèges nécessaires, exécutez les commandes suivantes.

```
CREATE USER '<user acct>'@'%' IDENTIFIED BY '<user password>';
GRANT ALTER, CREATE, DROP, INDEX, INSERT, UPDATE, DELETE, SELECT ON <schema>.* TO
'<user acct>'@'%';
GRANT ALL PRIVILEGES ON awsdms_control.* TO '<user acct>'@'%';
```

- Pendant la phase de migration de chargement complet, vous devez désactiver les clés étrangères sur vos tables cibles. Pour désactiver les contrôles de clé étrangère sur une base de données compatible MySQL pendant un chargement complet, vous pouvez ajouter la commande suivante à la section Attributs de connexion supplémentaires de la console AWS DMS pour votre point de terminaison cible.

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

- Définissez le paramètre de base de données `local_infile = 1` pour permettre à AWS DMS de charger les données dans la base de données cible.

Limitations sur l'utilisation d'une base de données compatible MySQL comme cible pour AWS Database Migration Service

Lorsque vous utilisez une base de données MySQL comme cible, AWS DMS ne prend pas en charge les éléments suivants :

- Les instructions en langage de définition de données (DDL) TRUNCATE PARTITION, DROP TABLE et RENAME TABLE.
- Utilisation d'une instruction `ALTER TABLE` *`table_name`* `ADD COLUMN` *`column_name`* pour ajouter des colonnes au début ou au milieu d'une table.

- Lorsque vous chargez des données sur une cible compatible MySQL dans le cadre d'une tâche de chargement complet, AWS DMS ne signale pas les erreurs provoquées par des contraintes dans les journaux de tâches, ce qui peut entraîner des erreurs clés en double ou des incohérences avec le nombre d'enregistrements. Cela est dû à la façon dont MySQL gère les données locales avec la commande `LOAD DATA`. Veillez à effectuer les opérations suivantes pendant la phase de chargement complet :
  - Désactivez les contraintes.
  - Utilisez la validation AWS DMS pour vous assurer que les données sont cohérentes.
- Lorsque vous mettez à jour la valeur d'une colonne à sa valeur existante, les bases de données compatibles MySQL renvoient un avertissement `0 rows affected`. Bien que ce comportement ne soit pas techniquement une erreur, il est différent de la façon dont la situation est gérée par d'autres moteurs de base de données. Par exemple, Oracle effectue une mise à jour d'une seule ligne. Pour les bases de données compatibles MySQL, AWS DMS génère une entrée dans la table de contrôle awsdms_apply_exceptions et consigne l'avertissement suivant.

  ```
  Some changes from the source database had no impact when applied to
  the target database. See awsdms_apply_exceptions table for details.
  ```

- Aurora sans serveur est disponible en tant que cible pour Amazon Aurora version 1, compatible avec MySQL version 5.6. Aurora sans serveur est disponible en tant que cible pour Amazon Aurora version 2, compatible avec MySQL version 5.7. (Sélectionnez Aurora MySQL version 2.07.1 pour pouvoir utiliser Aurora sans serveur avec la compatibilité MySQL 5.7.) Pour plus d'informations sur Aurora sans serveur, consultez Utilisation d'Amazon Aurora sans serveur dans le Guide de l'utilisateur Amazon Aurora.
- AWS DMS ne prend pas en charge l'utilisation d'un point de terminaison de lecteur pour Aurora ou Amazon RDS, sauf si les instances sont en mode écriture, c'est-à-dire que les paramètres `read_only` et `innodb_read_only` sont définis sur `0` ou `OFF`. Pour plus d'informations sur l'utilisation d'Amazon RDS et Aurora en tant que cibles, consultez les rubriques suivantes :
  - Détermination de l'instance de base de données à laquelle vous êtes connecté
  - Mise à jour des réplicas en lecture avec MySQL

Paramètres de point de terminaison lors de l'utilisation d'une base de données compatible MySQL en tant que cible pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données cible compatible MySQL comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison cible à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--my-sql-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec MySQL en tant que cible sont indiqués dans le tableau suivant.

Name (Nom)	Description
`TargetDbType`	Spécifie l'endroit où migrer des tables sources sur la cible, vers une seule base de données ou plusieurs bases de données. Si vous spécifiez `SPECIFIC_DATABASE` , vous devez spécifier le nom de la base de données, lorsque vous utilisez AWS CLI ou AWS Management Console. Valeur par défaut : `MULTIPLE_DATABASES` Valeurs valides : {`SPECIFIC_DATABASE` , `MULTIPLE_DATABASES` } Exemple : `--my-sql-settings '{"Target DbType": "MULTIPLE_DATABASES"}'`
`ParallelLoadThreads`	Améliore les performances lors du chargement des données dans la base de données cible compatible MySQL. Spécifie le nombre de threads à utiliser pour charger les données dans la base de données cible compatible MySQL. La définition d'un grand nombre de threads peut avoir un impact négatif sur les performances de base de données, dans la mesure où une connexion distincte est requise pour chaque thread. Valeur par défaut : 1

Name (Nom)	Description
	Valeurs valides : 1 à 5 Exemple : `--my-sql-settings '{"ParallelLoadThreads": 1}'`
`AfterConnectScript`	Spécifie un script à exécuter immédiatement après la connexion de AWS DMS au point de terminaison. Par exemple, vous pouvez indiquer que la cible compatible MySQL doit traduire les instructions reçues dans le jeu de caractères latin1, qui est le jeu de caractères compilé par défaut de la base de données. Ce paramètre améliore généralement les performances lors d'une conversion à partir de clients UTF8. Exemple : `--my-sql-settings '{"AfterConnectScript": "SET character_set_connection='latin1'"}'`
`MaxFileSize`	Spécifie la taille maximale (en Ko) de tout fichier .csv utilisé pour transférer des données vers une base de données compatible avec MySQL. Valeur par défaut : 32 768 Ko (32 Mo) Valeurs valides : 1 à 1 048 576 `--my-sql-settings '{"MaxFileSize": 512}'`

Name (Nom)	Description
`CleanSrcMetadataOnMismatch`	Nettoie et recrée les informations de métadonnées de table sur l'instance de réplication lorsqu'une non-corre spondance survient. Par exemple, lorsque l'exécution d'une instruction alter DDL sur la table pourrait générer des informations différentes sur la table mise en cache dans l'instance de réplication. Booléen. Valeur par défaut : `false` Exemple : `--my-sql-settings '{"CleanSrcMetadataOnMismatch": false}'`

Vous pouvez également utiliser les attributs de connexion supplémentaires pour configurer la base de données cible compatible MySQL.

Le tableau suivant indique les attributs de connexion supplémentaires que vous pouvez utiliser avec MySQL en tant que cible.

Name (Nom)	Description
`Initstmt=SET FOREIGN_KEY_CHECKS=0;`	Désactive les contrôles de clés étrangères. Exemple : `--extra-connection-attributes "Initstmt=SET FOREIGN_KEY_CHECKS=0;"`
`Initstmt=SET time_zone`	Spécifie le fuseau horaire pour la base de données compatible MySQL cible. Valeur par défaut : UTC Valeurs valides : noms des fuseaux horaires disponibles dans la base de données MySQL cible. Exemple : `--extra-connection-attributes "Initstmt=SET time_zone= US/Pacific ;"`

Vous pouvez également utiliser le paramètre `AfterConnectScript` de la commande `--my-sql-settings` pour désactiver les contrôles de clés étrangères et spécifier le fuseau horaire de la base de données.

Types de données cibles pour MySQL

Le tableau suivant présente les types de données cibles de bases de données MySQL qui sont pris en charge lorsque vous utilisez AWS DMS et le mapping par défaut des types de données AWS DMS.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Types de données AWS DMS	Types de données MySQL
BOOLEAN	BOOLEAN
BYTES	Si la longueur est comprise entre 1 et 65 535, utilisez VARBINARY (length). Si la longueur est comprise entre 65 536 et 2 147 483 647, utilisez LONGLOB.
DATE	DATE
TIME	TIME
TIMESTAMP	« Si l'échelle est => 0 et =< 6, alors : DATETIME (Scale) Si l'échelle est => 7 et =< 9, alors : VARCHAR (37) »
INT1	TINYINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT

Types de données AWS DMS	Types de données MySQL
NUMERIC	DECIMAL (p,s)
REAL4	FLOAT
REAL8	DOUBLE PRECISION
CHAÎNE	Si la longueur est comprise entre 1 et 21 845, utilisez VARCHAR (length). Si la longueur est comprise entre 21 846 et 2 147 483 647, utilisez LONGTEXT.
UINT1	UNSIGNED TINYINT
UINT2	UNSIGNED SMALLINT
UINT4	UNSIGNED INTEGER
UINT8	UNSIGNED BIGINT
WSTRING	Si la longueur est comprise entre 1 et 32 767, utilisez VARCHAR (length). Si la longueur est comprise entre 32 768 et 2 147 483 647, utilisez LONGTEXT.
BLOB	Si la longueur est comprise entre 1 et 65 535, utilisez BLOB. Si la longueur est comprise entre 65 536 et 2 147 483 647, utilisez LONGBLOB. Si la longueur est 0, utilisez LONGBLOB (full LOB support).

Types de données AWS DMS	Types de données MySQL
NCLOB	Si la longueur est comprise entre 1 et 65 535, utilisez TEXT. Si la longueur est comprise entre 65 536 et 2 147 483 647, utilisez LONGTEXT avec ucs2 pour CHARACTER SET. Si la longueur est 0, utilisez LONGTEXT (full LOB support) avec ucs2 pour CHARACTER SET.
CLOB	Si la longueur est comprise entre 1 et 65 535, utilisez TEXT. Si la longueur est comprise entre 65 536 et 2 147 483 647, utilisez LONGTEXT. Si la longueur est 0, utilisez LONGTEXT (full LOB support).

Utilisation d'une base de données Amazon Redshift en tant que cible pour AWS Database Migration Service

Vous pouvez migrer des données vers des bases de données Amazon Redshift à l'aide d'AWS Database Migration Service. Amazon Redshift est un service d'entreposage de données entièrement géré dans le cloud. Avec une base de données Amazon Redshift en tant que cible, vous pouvez migrer les données provenant de tous les autres bases de données sources prises en charge.

Vous pouvez utiliser Amazon Redshift sans serveur en tant que cible pour AWS DMS. Pour plus d'informations, consultez Utilisation d'AWS DMS avec Amazon Redshift sans serveur en tant que cible, ci-après.

Le cluster Amazon Redshift doit se trouver dans le même compte AWS et la même région AWS que l'instance de réplication.

Lors d'une migration de base de données vers Amazon Redshift, AWS DMS déplace d'abord les données dans un compartiment Amazon S3. Une fois que les fichiers se trouvent dans un

compartiment Amazon S3, AWS DMS les transfère vers les tables appropriés dans l'entrepôt des données Amazon Redshift. AWS DMS crée le compartiment S3 dans la même région AWS que la base de données Amazon Redshift. L'instance de réplication AWS DMS doit se trouver dans cette même région AWS.

Si vous utilisez AWS CLI ou l'API DMS pour migrer les données vers Amazon Redshift, configurez un rôle AWS Identity and Access Management (IAM) pour autoriser l'accès à S3. Pour plus d'informations sur la création de ce rôle IAM, consultez Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS.

Le point de terminaison Amazon Redshift automatise complètement les éléments suivants :

- la génération de schéma et le mappage de type de données ;
- le chargement complet des tables de base de données source ;
- le chargement incrémentiel des modifications apportées aux tables source ;
- l'application des modifications de schéma en langage de définition de données (DDL) apportées aux tables de la source ;
- la synchronisation entre le chargement complet et les traitements de capture des données modifiées (CDC).

AWS Database Migration Service prend en charge les opérations de chargement complet et de traitement des modifications. AWS DMS lit les données de la base de données source et crée une série de fichiers de valeurs séparées par des virgules (.csv). Pour les opérations de chargement complet, AWS DMS crée des fichiers pour chaque table. AWS DMS copie ensuite les fichiers de chaque table dans un dossier distinct dans Amazon S3. Lorsque les fichiers sont chargés dans Amazon S3, AWS DMS envoie une commande de copie et les données des fichiers sont copiées dans Amazon Redshift. Pour les opérations de traitement des modifications, AWS DMS copie les modifications nettes dans les fichiers .csv. AWS DMS charge ensuite les fichiers de modifications nettes sur Amazon S3 et copie les données dans Amazon Redshift.

Pour plus de détails sur l'utilisation d'Amazon Redshift en tant que cible pour AWS DMS, consultez les sections suivantes :

Rubriques

- Conditions requises pour l'utilisation d'une base de données Amazon Redshift en tant que cible pour AWS Database Migration Service
- Privilèges requis pour utiliser Redshift en tant que cible

- Limitations de l'utilisation d'Amazon Redshift en tant que cible pour AWS Database Migration Service
- Configuration d'une base de données Amazon Redshift en tant que cible pour AWS Database Migration Service
- Utilisation du routage VPC amélioré avec Amazon Redshift en tant que cible pour AWS Database Migration Service
- Création et utilisation de clés AWS KMS pour chiffrer les données cibles Amazon Redshift
- Paramètres de point de terminaison lors de l'utilisation d'Amazon Redshift en tant que cible pour AWS DMS
- Utilisation d'une clé de chiffrement des données et d'un compartiment Amazon S3 comme stockage intermédiaire
- Paramètres de tâche multithread pour Amazon Redshift
- Types de données cibles pour Amazon Redshift
- Utilisation d'AWS DMS avec Amazon Redshift sans serveur en tant que cible

Conditions requises pour l'utilisation d'une base de données Amazon Redshift en tant que cible pour AWS Database Migration Service

La liste suivante décrit les conditions requises pour l'utilisation d'Amazon Redshift en tant que cible pour la migration des données :

- Utilisez AWS Management Console pour lancer un cluster Amazon Redshift. Notez les informations de base concernant votre compte AWS et votre cluster Amazon Redshift, telles que votre mot de passe, nom d'utilisateur et le nom de base de données. Vous avez besoin de ces valeurs lors de la création du point de terminaison cible Amazon Redshift.
- Le cluster Amazon Redshift doit se trouver dans le même compte AWS et la même région AWS que l'instance de réplication.
- L'instance de réplication AWS DMS a besoin d'une connectivité réseau au point de terminaison Amazon Redshift (nom d'hôte et port) que votre cluster utilise.
- AWS DMS utilise un compartiment Amazon S3 pour transférer les données vers la base de données Amazon Redshift. Pour que AWS DMS crée le compartiment, la console doit utiliser un rôle IAM `dms-access-for-endpoint`. Si vous utilisez AWS CLI ou l'API DMS pour créer une migration de base de données avec Amazon Redshift en tant que base de données cible, vous

devez créer ce rôle IAM. Pour plus d'informations sur la création de ce rôle, consultez la page Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS.

- AWS DMS convertit les objets BLOB, CLOB et NCLOB en VARCHAR sur l'instance Amazon Redshift cible. Amazon Redshift ne prend pas en charge les types de données VARCHAR supérieurs à 64 Ko. Par conséquent, vous ne pouvez pas stocker les objets binaires volumineux (LOB) traditionnels sur Amazon Redshift.
- Définissez le paramètre de tâche de métadonnées cible BatchApplyEnabled sur `true` pour que AWS DMS gère les modifications apportées aux tables cibles Amazon Redshift pendant la CDC. Une clé primaire sur la table source et la table cible est requise. Sans clé primaire, les modifications sont appliquées instruction par instruction. Ce qui peut nuire aux performances des tâches pendant la CDC en provoquant une latence cible et en affectant la file d'attente de validation du cluster.

Privilèges requis pour utiliser Redshift en tant que cible

Utilisez la commande GRANT pour définir les privilèges d'accès pour un utilisateur ou un groupe d'utilisateurs. Les privilèges incluent les options d'accès telles que pouvoir lire les données des tables et des vues, écrire des données et créer des tables. Pour plus d'informations sur l'utilisation de GRANT avec Amazon Redshift, consultez GRANT dans le Guide du développeur de base de données Amazon Redshift.

La syntaxe suivante s'applique à accorder des privilèges spécifiques pour une table, une base de données, un schéma, une fonction, une procédure ou des privilèges de niveau langue sur les tables et les vues Amazon Redshift.

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | REFERENCES } [,...] | ALL
 [ PRIVILEGES ] }
    ON { [ TABLE ] table_name [, ...] | ALL TABLES IN SCHEMA schema_name [, ...] }
    TO { username [ WITH GRANT OPTION ] | GROUP group_name | PUBLIC } [, ...]

GRANT { { CREATE | TEMPORARY | TEMP } [,...] | ALL [ PRIVILEGES ] }
    ON DATABASE db_name [, ...]
    TO { username [ WITH GRANT OPTION ] | GROUP group_name | PUBLIC } [, ...]

GRANT { { CREATE | USAGE } [,...] | ALL [ PRIVILEGES ] }
    ON SCHEMA schema_name [, ...]
    TO { username [ WITH GRANT OPTION ] | GROUP group_name | PUBLIC } [, ...]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }
```

```
    ON { FUNCTION function_name ( [ [ argname ] argtype [, ...] ] ) [, ...] | ALL
 FUNCTIONS IN SCHEMA schema_name [, ...] }
    TO { username [ WITH GRANT OPTION ] | GROUP group_name | PUBLIC } [, ...]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }
    ON { PROCEDURE procedure_name ( [ [ argname ] argtype [, ...] ] ) [, ...] | ALL
 PROCEDURES IN SCHEMA schema_name [, ...] }
    TO { username [ WITH GRANT OPTION ] | GROUP group_name | PUBLIC } [, ...]

GRANT USAGE
    ON LANGUAGE language_name [, ...]
    TO { username [ WITH GRANT OPTION ] | GROUP group_name | PUBLIC } [, ...]
```

La syntaxe suivante s'applique aux privilèges de niveau colonne sur les tables et les vues Amazon Redshift.

```
GRANT { { SELECT | UPDATE } ( column_name [, ...] ) [, ...] | ALL [ PRIVILEGES ]
 ( column_name [,...] ) }
     ON { [ TABLE ] table_name [, ...] }
     TO { username | GROUP group_name | PUBLIC } [, ...]
```

Voici la syntaxe du privilège ASSUMEROLE accordé aux utilisateurs et aux groupes ayant un rôle spécifié.

```
GRANT ASSUMEROLE
    ON { 'iam_role' [, ...] | ALL }
    TO { username | GROUP group_name | PUBLIC } [, ...]
    FOR { ALL | COPY | UNLOAD } [, ...]
```

Limitations de l'utilisation d'Amazon Redshift en tant que cible pour AWS Database Migration Service

Les limitations suivantes s'appliquent lors de l'utilisation d'une base de données Amazon Redshift en tant que cible :

- N'activez pas la gestion des versions pour le compartiment S3 que vous utilisez comme stockage intermédiaire pour votre cible Amazon Redshift. Si vous avez besoin de la gestion des versions S3, utilisez des politiques de cycle de vie pour supprimer les anciennes versions de manière active. Dans le cas contraire, la connexion de test du point de terminaison risque d'échouer en raison du délai d'attente d'appel `list-object` S3. Pour créer une politique de cycle de vie pour un

compartiment S3, consultez Gestion du cycle de vie de votre stockage. Pour supprimer une version d'un objet S3, consultez Suppression des versions d'objet d'un compartiment activé pour la gestion des versions.

- La commande DDL suivante n'est pas prise en charge :

```
ALTER TABLE table name MODIFY COLUMN column name data type;
```

- AWS DMS ne peut pas migrer ou répliquer des modifications apportées à un schéma avec un nom qui commence par un caractère de soulignement (_). Si vous avez des schémas dont le nom commence par un trait de soulignement, utilisez les transformations de mappage pour renommer le schéma sur la cible.
- Amazon Redshift ne prend pas en charge les VARCHAR supérieurs à 64 Ko. Les objets binaires volumineux (LOB) provenant de bases de données traditionnelles ne peuvent pas être stockés dans Amazon Redshift.
- L'application d'une instruction DELETE à une table avec une clé primaire à plusieurs colonnes n'est pas prise en charge lorsque l'un des noms de colonne de clé primaire utilise un mot réservé. Pour obtenir la liste des mots réservés Amazon Redshift, consultez cette page.
- Vous pouvez rencontrer des problèmes de performances si votre système source exécute des opérations UPDATE sur la clé primaire d'une table source. Ces problèmes de performances se produisent lors de l'application de modifications à la cible. Cela est dû au fait que les opérations UPDATE (et DELETE) dépendent de la valeur de la clé primaire pour identifier la ligne cible. Si vous mettez à jour la clé primaire d'une table source, votre journal des tâches contiendra des messages tels que les suivants :

```
Update on table 1 changes PK to a PK that was previously updated in the same bulk
 update.
```

- DMS ne prend pas en charge les noms DNS personnalisés lors de la configuration d'un point de terminaison pour un cluster Redshift, et vous devez utiliser le nom DNS fourni par Amazon. Étant donné que le cluster Amazon Redshift doit se trouver dans le même compte AWS et dans la même région que l'instance de réplication, la validation échoue si vous utilisez un point de terminaison DNS personnalisé.
- Le délai d'expiration des sessions inactives par défaut d'Amazon Redshift est de 4 heures. Lorsqu'il n'y a aucune activité dans le cadre de la tâche de réplication DMS, Redshift déconnecte la session au bout de 4 heures. Des erreurs peuvent survenir si DMS ne parvient pas à se connecter et

doit éventuellement redémarrer. Pour contourner le problème, définissez une limite SESSION TIMEOUT supérieure à 4 heures pour l'utilisateur de réplication DMS. Vous pouvez également consulter la description de ALTER USER dans le Guide du développeur de base de données Amazon Redshift.

- Lorsqu'AWS DMS réplique des données de la table source sans clé primaire ou unique, la latence de la CDC peut être élevée, ce qui se traduit par un niveau de performance inacceptable.

Configuration d'une base de données Amazon Redshift en tant que cible pour AWS Database Migration Service

AWS Database Migration Service doit être configuré pour fonctionner avec l'instance Amazon Redshift. Le tableau suivant décrit les propriétés de configuration disponibles pour le point de terminaison Amazon Redshift.

Propriété	Description
serveur	Nom du cluster Amazon Redshift que vous utilisez.
port	Numéro de port d'Amazon Redshift. La valeur par défaut est 5439.
username	Nom d'utilisateur Amazon Redshift d'un utilisateur enregistré.
mot de passe	Mot de passe de l'utilisateur nommé dans la propriété de nom d'utilisateur.
database	Nom de l'entrepôt des données (service) Amazon Redshift que vous utilisez.

Si vous souhaitez ajouter des attributs de chaîne de connexion supplémentaires au point de terminaison Amazon Redshift, vous pouvez spécifier les attributs `maxFileSize` et `fileTransferUploadStreams`. Pour plus d'informations sur ces attributs, consultez la page Paramètres de point de terminaison lors de l'utilisation d'Amazon Redshift en tant que cible pour AWS DMS.

Utilisation du routage VPC amélioré avec Amazon Redshift en tant que cible pour AWS Database Migration Service

Si vous utilisez le routage VPC amélioré avec votre cible Amazon Redshift, l'ensemble du trafic COPY entre votre cluster Amazon Redshift et vos référentiels de données passe par votre VPC. Étant donné que le routage VPC amélioré affecte la manière dont Amazon Redshift accède aux autres ressources, les commandes COPY risquent d'échouer si vous n'avez pas configuré correctement votre VPC.

AWS DMS peut être affecté par ce comportement, car il utilise la commande COPY pour déplacer des données dans S3 vers un cluster Amazon Redshift.

Voici les étapes exécutées par AWS DMS pour charger des données dans une cible Amazon Redshift :

1. AWS DMS copie les données de la source vers des fichiers CSV sur le serveur de réplication.
2. AWS DMS utilise le kit SDK AWS pour copier les fichiers .csv dans un compartiment S3 sur votre compte.
3. AWS DMS utilise ensuite la commande COPY dans Amazon Redshift pour copier les données depuis les fichiers .csv dans S3 vers une table appropriée dans Amazon Redshift.

Si le routage VPC amélioré n'est pas activé, Amazon Redshift route le trafic via Internet, y compris le trafic vers d'autres services au sein du réseau AWS. Si la fonction n'est pas activée, vous n'avez pas à configurer le chemin d'accès réseau. Si la fonction est activée, vous devez créer spécifiquement un chemin d'accès réseau entre le VPC de votre cluster et vos ressources de données. Pour plus d'informations sur la configuration requise, consultez Routage VPC amélioré dans la documentation d'Amazon Redshift.

Création et utilisation de clés AWS KMS pour chiffrer les données cibles Amazon Redshift

Vous pouvez chiffrer vos données cibles transmises à Amazon S3 avant de les copier sur Amazon Redshift. Pour ce faire, vous pouvez créer et utiliser des clés AWS KMS personnalisées. Vous pouvez utiliser la clé que vous avez créée pour chiffrer vos données cibles à l'aide de l'un des mécanismes suivants lorsque vous créez le point de terminaison cible Amazon Redshift :

- Utilisez l'option suivante lorsque vous exécutez la commande `create-endpoint` à l'aide de l'AWS CLI.

```
--redshift-settings '{"EncryptionMode": "SSE_KMS", "ServerSideEncryptionKmsKeyId":
 "your-kms-key-ARN"}'
```

Ici, *your-kms-key-ARN* est l'Amazon Resource Name (ARN) pour votre clé KMS. Pour de plus amples informations, veuillez consulter Utilisation d'une clé de chiffrement des données et d'un compartiment Amazon S3 comme stockage intermédiaire.

- Définissez l'attribut de connexion supplémentaire `encryptionMode` à la valeur `SSE_KMS` et l'attribut de connexion supplémentaire `serverSideEncryptionKmsKeyId` à l'ARN de votre clé KMS. Pour de plus amples informations, veuillez consulter Paramètres de point de terminaison lors de l'utilisation d'Amazon Redshift en tant que cible pour AWS DMS.

Pour chiffrer des données cibles Amazon Redshift à l'aide d'une clé KMS, vous avez besoin d'un rôle AWS Identity and Access Management (IAM) qui dispose des autorisations pour accéder aux données Amazon Redshift. Ce rôle IAM est ensuite accessible dans une stratégie (stratégie de clé) attachée à la clé de chiffrement que vous créez. Pour ce faire, créez les éléments suivants dans votre console IAM :

- Un rôle IAM avec une stratégie gérée par AWS.
- Une clé KMS avec une stratégie de clé qui fait référence à ce rôle.

Les procédures suivantes décrivent la marche à suivre.

Pour créer un rôle IAM avec la stratégie gérée par AWS requise

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, sélectionnez Rôles. La page Rôles s'ouvre.
3. Sélectionnez Create role (Créer un rôle). La page Créer un rôle s'ouvre.
4. Choisissez Service AWS en tant qu'entité de confiance, puis DMS en tant que service utilisant le rôle.
5. Sélectionnez Next: Permissions (Étape suivante : autorisations). La page Attach permissions policies (Attacher des stratégies d'autorisation) s'affiche.
6. Recherchez et sélectionnez la stratégie `AmazonDMSRedshiftS3Role`.
7. Choisissez Suivant : Balises. La page Ajouter des balises s'affiche. Ici, vous pouvez ajouter toutes les balises souhaitées.

8. Choisissez Next: Review (Suivant : Vérification) et vérifiez vos résultats.
9. Si les paramètres correspondent à vos besoins, saisissez un nom pour le rôle (par exemple, `DMS-Redshift-endpoint-access-role`) et une description supplémentaire si vous le souhaitez, puis choisissez Créer un rôle. La page Rôles s'ouvre avec un message indiquant que votre rôle a été créé.

Vous avez désormais créé le nouveau rôle pour accéder aux ressources Amazon Redshift en vue du chiffrement avec un nom spécifié, par exemple `DMS-Redshift-endpoint-access-role`.

Pour créer une clé de chiffrement AWS KMS avec une stratégie de clé qui fait référence à votre rôle IAM.

> **Note**
>
> Pour plus d'informations sur le fonctionnement d'AWS DMS avec les clés de chiffrement AWS KMS, consultez Définition d'une clé de chiffrement et spécification des autorisations AWS KMS.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Choisissez Create key. La page Configurer la clé s'ouvre.
5. Pour Type de clé, choisissez Symétrique.

   > **Note**
   >
   > Vous ne pouvez créer qu'une clé symétrique, car tous les services AWS, tels qu'Amazon Redshift, ne fonctionnent qu'avec des clés de chiffrement symétriques.

6. Choisissez Options avancées. Pour Origine des éléments de clé, assurez-vous que KMS est choisi, puis choisissez Suivant. La page Ajouter des étiquettes s'ouvre.
7. Dans Créer un alias et une description, entrez un alias pour la clé (par exemple, `DMS-Redshift-endpoint-encryption-key`) et toute description supplémentaire.

8. Pour Balises, ajoutez les balises que vous souhaitez pour vous aider à identifier la clé et suivre son utilisation, puis choisissez Suivant. La page Définir des autorisations d'administration de clé s'ouvre et affiche une liste d'utilisateurs et de rôles parmi lesquels vous pouvez choisir.
9. Ajoutez les utilisateurs et les rôles que vous souhaitez voir gérer la clé. Assurez-vous que ces utilisateurs et ces rôles ont les autorisations requises pour gérer la clé.
10. Pour Suppression de clé, choisissez si les administrateurs de clé peuvent supprimer celle-ci, puis choisissez Suivant. La page Définir des autorisations d'utilisation de clé s'ouvre et affiche une liste supplémentaire d'utilisateurs et de rôles parmi lesquels vous pouvez choisir.
11. Pour Ce compte, choisissez les utilisateurs disponibles pour lesquels vous souhaitez effectuer des opérations de chiffrement sur des cibles Amazon Redshift. Choisissez également le rôle que vous avez créé précédemment dans Rôles pour activer l'accès au chiffrement des objets cibles Amazon Redshift, par exemple `DMS-Redshift-endpoint-access-role`.
12. Si vous souhaitez ajouter d'autres comptes non répertoriés pour avoir ce même accès, pour Autres comptes AWS, choisissez Ajouter un autre compte AWS, puis Suivant. La page Réviser et modifier la stratégie de clé s'ouvre, affichant le JSON de la stratégie de clé que vous pouvez réviser et modifier en saisissant les informations dans le JSON existant. Ici, vous pouvez voir où la stratégie clé fait référence au rôle et aux utilisateurs (par exemple, `Admin` et `User1`) que vous avez choisis à l'étape précédente. Vous pouvez également voir les actions clés autorisées pour les différents mandataires (utilisateurs et rôles), comme illustré dans l'exemple suivant.

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:root"
        ]
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow access for Key Administrators",
      "Effect": "Allow",
      "Principal": {
```

```
        "AWS": [
          "arn:aws:iam::111122223333:role/Admin"
        ]
      },
      "Action": [
        "kms:Create*",
        "kms:Describe*",
        "kms:Enable*",
        "kms:List*",
        "kms:Put*",
        "kms:Update*",
        "kms:Revoke*",
        "kms:Disable*",
        "kms:Get*",
        "kms:Delete*",
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ScheduleKeyDeletion",
        "kms:CancelKeyDeletion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/DMS-Redshift-endpoint-access-role",
          "arn:aws:iam::111122223333:role/Admin",
          "arn:aws:iam::111122223333:role/User1"
        ]
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow attachment of persistent resources",
      "Effect": "Allow",
```

```
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/DMS-Redshift-endpoint-access-role",
          "arn:aws:iam::111122223333:role/Admin",
          "arn:aws:iam::111122223333:role/User1"
        ]
      },
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": "*",
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": true
        }
      }
    }
  ]
```

13. Choisissez Finish (Terminer). La page Clés de chiffrement s'ouvre et affiche un message indiquant que votre AWS KMS key a été créée.

Vous venez de créer une nouvelle clé KMS avec un alias spécifié (par exemple, `DMS-Redshift-endpoint-encryption-key`). Cette clé permet à AWS DMS de chiffrer les données cibles Amazon Redshift.

Paramètres de point de terminaison lors de l'utilisation d'Amazon Redshift en tant que cible pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données cible Amazon Redshift comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison cible à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--redshift-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec Amazon Redshift en tant que cible sont indiqués dans le tableau suivant.

Name (Nom)	Description
`MaxFileSize`	Spécifie la taille maximale (en ko) de tout fichier .csv utilisé pour transférer des données vers Amazon Redshift. Valeur par défaut : 32 768 Ko (32 Mo) Valeurs valides : 1 à 1 048 576 Exemple : `--redshift-settings '{"MaxFileSize": 512}'`
`FileTransferUploadStreams`	Spécifie le nombre de threads utilisés pour télécharger un seul fichier. Valeur par défaut : 10 Valeurs valides : 1 à 64 Exemple : `--redshift-settings '{"FileTransferUploadStreams": 20}'`
`Acceptanydate`	Spécifie si un format de date quelconque est accepté, y compris les formats de dates non valides tels que 0000-00-00. Valeur booléenne. Valeur par défaut : false Valeurs valides : true | false Exemple : `--redshift-settings '{"Acceptanydate": true}'`
`Dateformat`	Spécifie le format de date. Il s'agit d'une entrée de chaîne qui est vide par défaut. Le format par défaut est JJ-MM-DD, mais vous pouvez le remplacer par AAAA-MM-JJ, par exemple. Si vos valeurs de date ou d'heure utilisent des formats différents, utilisez l'argument `auto` avec le paramètre `Dateformat` . L'argument `auto` reconnaît

Name (Nom)	Description
	plusieurs formats qui ne sont pas pris en charge quand une chaîne `Dateformat` est utilisée. Le mot clé `auto` est sensible à la casse. Valeur par défaut : vide Valeurs valides : "*dateformat_string* " ou `auto` Exemple : `--redshift-settings '{"Datefo rmat": "auto"}'`
`Timeformat`	Spécifie le format de l'heure. Il s'agit d'une entrée de chaîne qui est vide par défaut. L'argument `auto` reconnaît plusieurs formats qui ne sont pas pris en charge quand une chaîne `Timeformat` est utilisée. Si vos valeurs de date et d'heure utilisent des formats différents les uns des autres, utilisez l'argument `auto` avec le paramètre `Timeformat` . Valeur par défaut : 10 Valeurs valides : "*Timeformat_string* " | "auto" | "epochsecs" | "epochmillisecs" Exemple : `--redshift-settings '{"Timefo rmat": "auto"}'`
`Emptyasnull`	Spécifie si AWS DMS doit migrer les champs CHAR et VARCHAR vides en tant que valeurs NULL. La valeur true définit les champs CHAR et VARCHAR vides comme étant null. Valeur par défaut : false Valeurs valides : true | false Exemple : `--redshift-settings '{"Emptya snull": true}'`

Name (Nom)	Description
`TruncateColumns`	Tronque les données des colonnes avec le nombre de caractères donné afin qu'il corresponde à la spécification de la colonne. S'applique uniquement aux colonnes avec un type de données VARCHAR ou CHAR et des lignes de 4 Mo ou moins. Valeur par défaut : false Valeurs valides : true | false Exemple : `--redshift-settings '{"TruncateColumns": true}'`
`RemoveQuotes`	Supprime les guillemets des chaînes dans les données entrantes. Tous les caractères compris entre les guillemets, y compris les délimiteurs, sont conservés. Pour plus d'informations sur la suppression des guillemets pour une cible Amazon Redshift, consultez le Manuel du développeur de base de données Amazon Redshift. Valeur par défaut : false Valeurs valides : true | false Exemple : `--redshift-settings '{"RemoveQuotes": true}'`
`TrimBlanks`	Supprime les caractères d'espace vide de fin d'une chaîne VARCHAR. Ce paramètre s'applique uniquement aux colonnes avec un type de données VARCHAR. Valeur par défaut : false Valeurs valides : true | false Exemple : `--redshift-settings '{"TrimBlanks": true}'`

Name (Nom)	Description
`EncryptionMode`	Le mode de chiffrement côté serveur que vous souhaitez utiliser pour transférer vos données vers S3 avant qu'elles ne soient copiées vers Amazon Redshift. Les valeurs valides sont `SSE_S3` (chiffrement côté serveur S3) ou `SSE_KMS` (chiffrement de clé KMS). Si vous choisissez `SSE_KMS`, définissez le paramètre `ServerSideEncryptionKmsKeyId` sur l'Amazon Resource Name (ARN) de la clé KMS à utiliser pour le chiffrement. **Note** Vous pouvez également utiliser la commande CLI `modify-endpoint` pour modifier la valeur du paramètre `EncryptionMode` d'un point de terminaison existant de `SSE_KMS` à `SSE_S3`. Vous ne pouvez pas modifier la valeur `EncryptionMode` de `SSE_S3` à `SSE_KMS`. Valeur par défaut : `SSE_S3` Valeurs valides : `SSE_S3` ou `SSE_KMS` Exemple : `--redshift-settings '{"EncryptionMode": "SSE_S3"}'`

Name (Nom)	Description
`ServerSideEncryptionKmsKeyId`	Si vous définissez `EncryptionMode` sur `SSE_KMS`, définissez ce paramètre sur l'ARN de la clé KMS. Vous pouvez rechercher cet ARN en sélectionnant l'alias de clé dans la liste des clés AWS KMS créées pour votre compte. Lorsque vous créez la clé, vous devez lui associer des stratégies et des rôles spécifiques. Pour de plus amples informations, veuillez consulter Création et utilisation de clés AWS KMS pour chiffrer les données cibles Amazon Redshift. Exemple : `--redshift-settings '{"ServerSideEncryptionKmsKeyId":"arn:aws:kms:us-east-1:111122223333:key/11a1a1a1-aaaa-9999-abab-2bbbbbb222a2"}'`
`EnableParallelBatchInMemoryCSVFiles`	Le paramètre `EnableParallelBatchInMemoryCSVFiles` améliore les performances des tâches de chargement complet multithread de grande taille en permettant à DMS d'écrire sur le disque plutôt que dans la mémoire. La valeur par défaut est `false`.
`CompressCsvFiles`	Utilisez cet attribut pour compresser les données envoyées à une cible Amazon Redshift pendant la migration. La valeur par défaut est `true` et la compression est activée par défaut.

Utilisation d'une clé de chiffrement des données et d'un compartiment Amazon S3 comme stockage intermédiaire

Vous pouvez utiliser les paramètres de point de terminaison cible Amazon Redshift pour configurer les éléments suivants :

- Une clé de chiffrement de données AWS KMS personnalisée. Vous pouvez ensuite utiliser cette clé pour chiffrer vos données transmises à Amazon S3 avant qu'elles ne soient copiées sur Amazon Redshift.

- Un compartiment S3 personnalisé en tant que stockage intermédiaire pour les données migrées vers Amazon Redshift.
- Mappez un booléen en tant que booléen provenant d'une source PostgreSQL. Par défaut, un type BOOLEAN est migré au format varchar(1). Vous pouvez indiquer à `MapBooleanAsBoolean` d'autoriser votre cible Redshift à migrer le type booléen en tant que booléen, comme illustré dans l'exemple suivant.

  ```
  --redshift-settings '{"MapBooleanAsBoolean": true}'
  ```

  Notez que vous devez définir ce paramètre à la fois sur les points de terminaison sources et cibles pour qu'il prenne effet.

Paramètres de clé KMS pour le chiffrement des données

Les exemples suivants illustrent la configuration d'une clé KMS personnalisée pour chiffrer vos données transmises à S3. Pour commencer, vous pouvez effectuer l'appel `create-endpoint` suivant à l'aide de l'AWS CLI.

```
aws dms create-endpoint --endpoint-identifier redshift-target-endpoint --endpoint-type
 target
--engine-name redshift --username your-username --password your-password
--server-name your-server-name --port 5439 --database-name your-db-name
--redshift-settings '{"EncryptionMode": "SSE_KMS",
"ServerSideEncryptionKmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/24c3c5a1-
f34a-4519-a85b-2debbef226d1"}'
```

Ici, l'objet JSON spécifié par l'option `--redshift-settings` définit deux paramètres. Le premier est le paramètre `EncryptionMode` avec la valeur `SSE_KMS`. L'autre est le paramètre `ServerSideEncryptionKmsKeyId` avec la valeur `arn:aws:kms:us-east-1:111122223333:key/24c3c5a1-f34a-4519-a85b-2debbef226d1`. Cette valeur est un Amazon Resource Name (ARN) pour votre clé KMS personnalisée.

Par défaut, le chiffrement de données S3 a lieu à l'aide du chiffrement côté serveur S3. Pour la cible Amazon Redshift de l'exemple précédent, cela équivaut à spécifier ses paramètres de point de terminaison, comme dans l'exemple suivant.

```
aws dms create-endpoint --endpoint-identifier redshift-target-endpoint --endpoint-type
 target
```

```
--engine-name redshift --username your-username --password your-password
--server-name your-server-name --port 5439 --database-name your-db-name
--redshift-settings '{"EncryptionMode": "SSE_S3"}'
```

Pour plus d'informations sur l'utilisation du chiffrement côté serveur S3, consultez Protection des données à l'aide du chiffrement côté serveur dans le Guide de l'utilisateur Amazon Simple Storage Service.

> **Note**
>
> Vous pouvez également utiliser la commande CLI `modify-endpoint` pour modifier la valeur du paramètre `EncryptionMode` d'un point de terminaison existant de `SSE_KMS` à `SSE_S3`. Vous ne pouvez pas modifier la valeur `EncryptionMode` de `SSE_S3` à `SSE_KMS`.

Paramètres du compartiment Amazon S3

Lorsque vous migrez des données vers un point de terminaison cible Amazon Redshift, AWS DMS utilise un compartiment Amazon S3 par défaut comme stockage de tâche intermédiaire avant de copier les données migrées dans Amazon Redshift. Par exemple, les exemples indiqués pour la création d'un point de terminaison cible Amazon Redshift avec une clé de chiffrement des données AWS KMS utilisent ce compartiment S3 par défaut (voir Paramètres de clé KMS pour le chiffrement des données).

Vous pouvez plutôt spécifier un compartiment S3 pour ce stockage intermédiaire en incluant les paramètres suivants dans la valeur de votre option `--redshift-settings` dans la commande `create-endpoint` de l'AWS CLI :

- `BucketName` : chaîne que vous spécifiez comme le nom de stockage du compartiment S3. Si votre rôle d'accès au service est basé sur la politique `AmazonDMSRedshiftS3Role`, cette valeur doit avoir le préfixe `dms-`, par exemple, `dms-my-bucket-name`.
- `BucketFolder` : (Facultatif) chaîne que vous pouvez spécifier comme nom du dossier de stockage dans le compartiment S3 spécifié.
- `ServiceAccessRoleArn` : ARN d'un rôle IAM qui autorise un accès administratif au compartiment S3. Généralement, vous créez ce rôle en fonction de la stratégie `AmazonDMSRedshiftS3Role`. Pour obtenir un exemple, consultez la procédure permettant de créer un rôle IAM avec la stratégie gérée par AWS requise dans Création et utilisation de clés AWS KMS pour chiffrer les données cibles Amazon Redshift.

> **Note**
>
> Si vous spécifiez l'ARN d'un autre rôle IAM à l'aide de l'option `--service-access-role-arn` de la commande `create-endpoint`, cette option de rôle IAM est prioritaire.

L'exemple suivant montre comment utiliser ces paramètres pour spécifier un compartiment Amazon S3 personnalisé dans l'appel `create-endpoint` suivant à l'aide d'AWS CLI.

```
aws dms create-endpoint --endpoint-identifier redshift-target-endpoint --endpoint-type
 target
--engine-name redshift --username your-username --password your-password
--server-name your-server-name --port 5439 --database-name your-db-name
--redshift-settings '{"ServiceAccessRoleArn": "your-service-access-ARN",
"BucketName": "your-bucket-name", "BucketFolder": "your-bucket-folder-name"}'
```

Paramètres de tâche multithread pour Amazon Redshift

Vous pouvez améliorer les performances des tâches de chargement complet et de capture des données de modification (CDC) pour un point de terminaison cible Amazon Redshift en utilisant des paramètres de tâche multithread. Ils vous permettent de spécifier le nombre de threads simultanés et le nombre d'enregistrements à stocker dans un tampon.

Paramètres de tâche de chargement complet multithread pour Amazon Redshift

Pour améliorer les performances de chargement complet, vous pouvez utiliser les paramètres de tâche `ParallelLoad*` suivants :

- `ParallelLoadThreads` : spécifie le nombre de threads simultanés utilisés par DMS pendant un chargement complet pour transférer des enregistrements de données vers un point de terminaison cible Amazon Redshift. La valeur par défaut est zéro (0) et la valeur maximale est 32. Pour de plus amples informations, veuillez consulter Paramètres de tâche de chargement complet.

  Définissez l'attribut `enableParallelBatchInMemoryCSVFiles` sur `false` lorsque vous utilisez le paramètre de tâche `ParallelLoadThreads`. L'attribut améliore les performances des tâches de chargement complet multithread de grande taille en permettant à DMS d'écrire sur le disque plutôt que dans la mémoire. La valeur par défaut est `true`.
- `ParallelLoadBufferSize` : spécifie le nombre maximal de demandes d'enregistrement de données lors de l'utilisation de threads de chargement parallèle avec la cible Redshift. La valeur

par défaut est 100 et la valeur maximale est 1 000. Nous vous recommandons d'utiliser cette option lorsque ParallelLoadThreads > 1 (supérieur à un).

> **Note**
>
> La prise en charge de l'utilisation des paramètres de tâche `ParallelLoad*` pendant la tâche FULL LOAD vers des points de terminaison cibles Amazon Redshift est disponible dans AWS DMS versions 3.4.5 et ultérieures.
> Le paramètre de point de terminaison Redshift `ReplaceInvalidChars` n'est pas pris en charge pour être utilisé lors de la capture des données de modification (CDC) ou lors d'une tâche de migration FULL LOAD compatible avec le chargement parallèle. Il est pris en charge pour la migration FULL LOAD lorsque le chargement parallèle n'est pas activé. Pour plus d'informations, consultez RedShiftSettings dans la Référence des API AWS Database Migration Service.

Paramètres de tâche de CDC multithread pour Amazon Redshift

Pour améliorer les performances de la CDC, vous pouvez utiliser les paramètres de tâche `ParallelApply*` suivants :

- `ParallelApplyThreads` : spécifie le nombre de threads simultanés utilisés par AWS DMS pendant un chargement CDC pour transférer des enregistrements de données vers un point de terminaison cible Amazon Redshift. La valeur par défaut est zéro (0) et la valeur maximale est 32. La valeur minimale recommandée est le nombre de tranches présentes dans votre cluster.
- `ParallelApplyBufferSize` : spécifie le nombre maximal de demandes d'enregistrement de données lors de l'utilisation de threads d'application parallèle avec la cible Redshift. La valeur par défaut est 100 et la valeur maximale est 1 000. Nous recommandons d'utiliser cette option lorsque ParallelApplyThreads > 1 (supérieur à un).

  Pour tirer le meilleur parti de Redshift en tant que cible, nous recommandons que la valeur de `ParallelApplyBufferSize` soit au moins deux fois supérieure à la valeur de `ParallelApplyThreads`.

> **Note**
>
> La prise en charge de l'utilisation des paramètres de tâche `ParallelApply*` pendant la CDC vers des points de terminaison cibles Amazon Redshift est disponible dans AWS DMS versions 3.4.3 et ultérieures.

Le niveau de parallélisme appliqué dépend de la corrélation entre la taille de lot totale et la taille de fichier maximale utilisées pour transférer les données. Lorsque vous utilisez des paramètres de tâche de CDC multithread avec une cible Redshift, vous tirez le meilleur parti lorsque la taille de lot est supérieure à la taille de fichier maximale. Par exemple, vous pouvez utiliser la combinaison suivante de paramètres de point de terminaison et de tâche pour optimiser les performances.

```
// Redshift endpoint setting

        MaxFileSize=250000;

// Task settings

        BatchApplyEnabled=true;
        BatchSplitSize =8000;
        BatchApplyTimeoutMax =1800;
        BatchApplyTimeoutMin =1800;
        ParallelApplyThreads=32;
        ParallelApplyBufferSize=100;
```

En utilisant les paramètres de l'exemple précédent, un client ayant une charge de travail transactionnelle importante tire le meilleur parti de sa mémoire tampon par lots, qui contient 8 000 enregistrements, est remplie en 1 800 secondes et utilise 32 threads parallèles avec une taille de fichier maximale de 250 Mo.

Pour de plus amples informations, veuillez consulter Paramètres de réglage du traitement des modifications.

> **Note**
>
> Les requêtes DMS exécutées pendant la réplication continue vers un cluster Redshift peuvent partager la même file d'attente WLM (gestion de la charge de travail) avec les autres requêtes d'application en cours d'exécution. Pensez donc à configurer correctement les

propriétés WLM afin d'influencer les performances lors de la réplication continue vers une cible Redshift. Par exemple, si d'autres requêtes ETL parallèles sont en cours d'exécution, DMS s'exécute plus lentement et les gains de performances sont perdus.

Types de données cibles pour Amazon Redshift

Le point de terminaison Amazon Redshift pour AWS DMS prend en charge la plupart des types de données Amazon Redshift. Le tableau suivant présente les types de données cibles Amazon Redshift qui sont pris en charge lorsque vous utilisez AWS DMS et le mappage par défaut à partir des types de données AWS DMS.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Types de données AWS DMS	Types de données Amazon Redshift
BOOLEAN	BOOL
BYTES	VARCHAR (Length)
DATE	DATE
TIME	VARCHAR(20)
DATETIME	Si l'échelle est => 0 et =< 6, selon le type de colonne cible Redshift, alors : TIMESTAMP (s) TIMESTAMPTZ (s) : si l'horodatage source contient un décalage de zone (comme dans SQL Server ou Oracle), il est converti en UTC lors de l'insertion/de la mise à jour. S'il ne contient pas de décalage, l'heure est déjà prise en compte en UTC. Si l'échelle est => 7 et =< 9, alors : VARCHAR (37)

Types de données AWS DMS	Types de données Amazon Redshift
INT1	INT2
INT2	INT2
INT4	INT4
INT8	INT8
NUMERIC	Si l'échelle est => 0 et =< 37, alors : NUMERIC (p,s) Si l'échelle est => 38 et =< 127, alors : VARCHAR (Length)
REAL4	FLOAT4
REAL8	FLOAT8
CHAÎNE	Si la longueur est comprise entre 1 et 65 535, utilisez VARCHAR (longueur en octets) Si la longueur est comprise entre 65 536 et 2 147 483 647, utilisez VARCHAR (65535)
UINT1	INT2
UINT2	INT2
UINT4	INT4
UINT8	NUMERIC (20,0)
WSTRING	Si la longueur est comprise entre 1 et 65 535, utilisez NVARCHAR (longueur en octets) Si la longueur est comprise entre 65 536 et 2 147 483 647, utilisez NVARCHAR (65535)

Types de données AWS DMS	Types de données Amazon Redshift
BLOB	VARCHAR (taille de LOB maximale *2) La taille de LOB maximale ne peut pas dépasser 31 Ko. Amazon Redshift ne prend pas en charge les VARCHAR supérieurs à 64 Ko.
NCLOB	NVARCHAR (taille de LOB maximale) La taille de LOB maximale ne peut pas dépasser 63 Ko. Amazon Redshift ne prend pas en charge les VARCHAR supérieurs à 64 Ko.
CLOB	VARCHAR (taille de LOB maximale) La taille de LOB maximale ne peut pas dépasser 63 Ko. Amazon Redshift ne prend pas en charge les VARCHAR supérieurs à 64 Ko.

Utilisation d'AWS DMS avec Amazon Redshift sans serveur en tant que cible

AWS DMS prend en charge l'utilisation d'Amazon Redshift sans serveur en tant que point de terminaison cible. Pour en savoir plus sur l'utilisation d'Amazon Redshift sans serveur, consultez Amazon Redshift sans serveur dans le Guide de gestion Amazon Redshift.

Cette rubrique explique comment utiliser un point de terminaison Amazon Redshift sans serveur avec AWS DMS.

> **Note**
>
> Lorsque vous créez un point de terminaison Amazon Redshift sans serveur, pour le champ DatabaseName de la configuration du point de terminaison RedshiftSettings, utilisez soit le nom de l'entrepôt des données Amazon Redshift, soit le nom du point de terminaison de groupe de travail. Pour le champ ServerName, utilisez la valeur du point de terminaison affichée sur la page Groupe de travail pour le cluster sans serveur (par exemple, `default-`

`workgroup.093291321484.us-east-1.redshift-serverless.amazonaws.com`). Pour en savoir plus sur la création d'un point de terminaison, consultez Création de points de terminaison source et cible. Pour en savoir plus sur le point de terminaison de groupe de travail, consultez Connexion à Amazon Redshift sans serveur.

Politique d'approbation avec Amazon Redshift sans serveur en tant que cible

Lorsque vous utilisez Amazon Redshift sans serveur en tant que point de terminaison cible, vous devez ajouter la section surlignée suivante à votre politique d'approbation. Cette politique d'approbation est liée au rôle `dms-access-for-endpoint`.

```
{
    "PolicyVersion": {
        "CreateDate": "2016-05-23T16:29:57Z",
        "VersionId": "v3",
        "Document": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "ec2:CreateNetworkInterface",
                    "ec2:DescribeAvailabilityZones",
                    "ec2:DescribeInternetGateways",
                    "ec2:DescribeSecurityGroups",
                    "ec2:DescribeSubnets",
                    "ec2:DescribeVpcs",
                    "ec2:DeleteNetworkInterface",
                    "ec2:ModifyNetworkInterfaceAttribute"
                ],
                "Resource": "arn:aws:service:region:account:resourcetype/id",
                "Effect": "Allow"
            },
            {
                "Sid": "",
                "Effect": "Allow",
                "Principal": {
                    "Service": "redshift-serverless.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
```

```
    },
    "IsDefaultVersion": true
    }
}
```

Pour plus d'informations sur l'utilisation d'une politique d'approbation avec AWS DMS, consultez Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS.

Limitations lors de l'utilisation d'Amazon Redshift sans serveur en tant que cible

L'utilisation de Redshift sans serveur en tant que cible présente les limitations suivantes :

- AWS DMS prend uniquement en charge Amazon Redshift sans serveur en tant que point de terminaison dans les régions qui prennent en charge Amazon Redshift sans serveur. Pour en savoir plus sur les régions qui prennent en charge Amazon Redshift sans serveur, consultez API Redshift sans serveur dans la rubrique Points de terminaison et quotas Amazon Redshift de la Référence générale AWS.
- Lorsque vous utilisez le routage VPC amélioré, assurez-vous de créer un point de terminaison Amazon S3 dans le même VPC que votre cluster Redshift sans serveur ou Redshift provisionné. Pour de plus amples informations, veuillez consulter Utilisation du routage VPC amélioré avec Amazon Redshift en tant que cible pour AWS Database Migration Service.
- AWS DMS sans serveur ne prend pas en charge Amazon Redshift sans serveur en tant que cible.

Utilisation d'une base de données SAP ASE comme cible pour AWS Database Migration Service

Vous pouvez migrer des données vers des bases de données SAP Adaptive Server Enterprise (ASE) (anciennement Sybase) à l'aide d'AWS DMS ou à partir de l'une des sources de base de données prises en charge.

Pour en savoir plus sur les versions de SAP ASE qui sont prises en charge par AWS DMS en tant que cible, consultez Objectifs pour AWS DMS.

Conditions requises pour l'utilisation de SAP ASE comme cible pour AWS Database Migration Service

Avant de commencer à utiliser une base de données SAP ASE comme cible pour AWS DMS, assurez-vous que les conditions préalables suivantes sont remplies :

- Fournissez à l'utilisateur de AWS DMS un accès au compte SAP ASE. Cet utilisateur doit disposer de privilèges de lecture/écriture dans la base de données SAP ASE.
- Dans certains cas, vous pouvez répliquer vers SAP ASE version 15.7 installée sur une instance Amazon EC2 de Microsoft Windows configurée avec des caractères non latins (par exemple, chinois). Dans de tels cas, AWS DMS exige que SAP ASE 15.7 SP121 soit installée sur la machine SAP ASE cible.

Limitations lors de l'utilisation d'une base de données SAP ASE en tant que cible pour AWS DMS

Les limitations suivantes s'appliquent lors de l'utilisation d'une base de données SAP ASE en tant que cible pour AWS DMS :

- AWS DMS ne prend pas en charge les tables qui contiennent des champs comportant les types de données suivants. Les colonnes répliquées avec ces types de données s'affichent en tant que valeurs NULL.
  - Type défini par l'utilisateur (UDT)

Paramètres de point de terminaison lors de l'utilisation de SAP ASE en tant que cible pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données cible SAP ASE comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison cible à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--sybase-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec SAP ASE en tant que cible sont indiqués dans le tableau suivant.

Name (Nom)	Description
`Driver`	Définissez cet attribut si vous souhaitez utiliser TLS pour ASE versions 15.7 et ultérieures. Valeur par défaut : `Adaptive Server Enterprise`

Name (Nom)	Description
	Exemple : `driver=Adaptive Server Enterprise 16.03.06;` Valeurs valides : `Adaptive Server Enterprise 16.03.06`
`AdditionalConnectionProperties`	Autres paramètres de connexion ODBC que vous souhaitez spécifier.

Types de données cibles pour SAP ASE

Le tableau suivant présente les types de données cibles de bases de données SAP ASE qui sont pris en charge lorsque vous utilisez AWS DMS et le mapping par défaut des types de données AWS DMS.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Types de données AWS DMS	Type de données SAP ASE
BOOLEAN	BIT
BYTES	VARBINARY (Length)
DATE	DATE
TIME	TIME
TIMESTAMP	Si l'échelle est => 0 et =< 6, alors : BIGDATETIME Si l'échelle est => 7 et =< 9, alors : VARCHAR (37)
INT1	TINYINT
INT2	SMALLINT

Types de données AWS DMS	Type de données SAP ASE
INT4	INTEGER
INT8	BIGINT
NUMERIC	NUMERIC (p,s)
REAL4	REAL
REAL8	DOUBLE PRECISION
CHAÎNE	VARCHAR (Length)
UINT1	TINYINT
UINT2	UNSIGNED SMALLINT
UINT4	UNSIGNED INTEGER
UINT8	UNSIGNED BIGINT
WSTRING	VARCHAR (Length)
BLOB	IMAGE
CLOB	UNITEXT
NCLOB	TEXT

Utilisation d'Amazon S3 en tant que cible pour AWS Database Migration Service

Vous pouvez migrer des données vers Amazon S3 en utilisant AWS DMS depuis n'importe quelle source de base de données prise en charge. Lorsque vous utilisez Amazon S3 en tant que cible dans une tâche AWS DMS, les données de chargement complet et de capture des données de modification (CDC) sont écrites au format .csv (valeurs séparées par une virgule) par défaut. Pour plus d'options de stockage compact et de requêtes plus rapides, vous avez également la possibilité d'écrire les données au format Apache Parquet (.parquet).

AWS DMS nomme les fichiers créés lors d'un chargement complet en utilisant un compteur hexadécimal incrémentiel ; par exemple, LOAD00001.csv, LOAD00002..., LOAD00009, LOAD0000A, etc. pour les fichiers .csv. AWS DMS nomme les fichiers CDC en utilisant des horodatages ; par exemple, 20141029-1134010000.csv. Pour chaque table source contenant des enregistrements, AWS DMS crée un dossier dans le dossier cible spécifié (si la table source n'est pas vide). AWS DMS écrit tous les fichiers de chargement complet et de CDC dans le compartiment Amazon S3 spécifié. Vous pouvez contrôler la taille des fichiers créés par AWS DMS à l'aide du paramètre de point de terminaison MaxFileSize.

Le paramètre `bucketFolder` contient l'emplacement où les fichiers .csv ou .parquet sont stockés avant d'être téléchargés vers le compartiment S3. Avec les fichiers .csv, les données de la table sont stockées dans le compartiment S3 au format suivant (l'exemple s'applique à des fichiers de chargement complet).

```
database_schema_name/table_name/LOAD00000001.csv
database_schema_name/table_name/LOAD00000002.csv
...
database_schema_name/table_name/LOAD00000009.csv
database_schema_name/table_name/LOAD0000000A.csv
database_schema_name/table_name/LOAD0000000B.csv
...database_schema_name/table_name/LOAD0000000F.csv
database_schema_name/table_name/LOAD00000010.csv
...
```

Vous pouvez spécifier le délimiteur de colonne, le délimiteur de ligne et d'autres paramètres à l'aide des attributs de connexion supplémentaires. Pour obtenir plus d'informations sur les attributs de connexion supplémentaires, consultez Paramètres de point de terminaison lors de l'utilisation d'Amazon S3 en tant que cible pour AWS DMS à la fin de cette section.

Vous pouvez spécifier le propriétaire du compartiment et empêcher toute tricherie en utilisant le paramètre de point de terminaison Amazon S3 `ExpectedBucketOwner`, comme indiqué ci-dessous. Ensuite, lorsque vous soumettez une demande pour tester une connexion ou effectuer une migration, S3 compare l'ID de compte du propriétaire du compartiment au paramètre spécifié.

```
--s3-settings='{"ExpectedBucketOwner": "AWS_Account_ID"}'
```

Lorsque vous utilisez AWS DMS pour répliquer les modifications de données à l'aide d'une tâche de CDC, la première colonne du fichier de sortie .csv ou .parquet indique la manière dont les données de ligne ont été modifiées pour le fichier .csv suivant.

```
I,101,Smith,Bob,4-Jun-14,New York
U,101,Smith,Bob,8-Oct-15,Los Angeles
U,101,Smith,Bob,13-Mar-17,Dallas
D,101,Smith,Bob,13-Mar-17,Dallas
```

Pour cet exemple, supposons qu'il y ait une `EMPLOYEE` table dans la base de données source. AWS DMS écrit les données dans le fichier .csv ou .parquet en réponse aux événements suivants :

- Un nouvel employé (Bob Smith, ID employé 101) est engagé le 4 juin 2014 au bureau de New York. Dans le fichier .csv ou .parquet, la valeur `I` dans la première colonne indique qu'une nouvelle ligne a fait l'objet d'une opération `INSERT` dans la table EMPLOYEE de la base de données source.
- Le 8 octobre 2015, Bob est transféré au bureau de Los Angeles. Dans le fichier .csv ou .parquet, la valeur `U` indique que la ligne correspondante de la table EMPLOYEE a fait l'objet d'une opération `UPDATE` pour refléter le nouvel emplacement du bureau de Bob. Le reste reflète la ligne de la table EMPLOYEE telle qu'elle apparaît après l'opération `UPDATE`.
- Le 13 mars 2017, Bob Dallas est à nouveau transféré au bureau de Dallas. Dans le fichier .csv ou .parquet, la valeur `U` indique que cette ligne a fait de nouveau l'objet d'une opération `UPDATE`. Le reste reflète la ligne de la table EMPLOYEE telle qu'elle apparaît après l'opération `UPDATE`.
- Après avoir travaillé quelque temps à Dallas, Bob quitte l'entreprise. Dans le fichier .csv ou .parquet, la valeur `D` indique que la ligne a fait l'objet d'une opération `DELETE` dans la table source. Le reste reflète la ligne de la table EMPLOYEE telle qu'elle apparaît avant la suppression.

Notez que par défaut pour la CDC, AWS DMS stocke les modifications de ligne pour chaque table de base de données sans tenir compte de l'ordre des transactions. Si vous souhaitez stocker les modifications de ligne dans les fichiers de CDC en fonction de l'ordre des transactions, vous devez utiliser les paramètres de point de terminaison S3 pour le spécifier ainsi que le chemin du dossier dans lequel vous souhaitez que les fichiers de transaction CDC soient stockés sur la cible S3. Pour de plus amples informations, veuillez consulter Capture des données de modification (CDC) avec l'ordre des transactions sur la cible S3.

Pour contrôler la fréquence des écritures sur une cible Amazon S3 lors d'une tâche de réplication de données, vous pouvez configurer les attributs de connexion supplémentaires `cdcMaxBatchInterval` et `cdcMinFileSize`. Cela peut améliorer les performances lors de l'analyse des données sans aucune surcharge supplémentaire. Pour de plus amples informations,

veuillez consulter Paramètres de point de terminaison lors de l'utilisation d'Amazon S3 en tant que cible pour AWS DMS.

Rubriques

- Prérequis pour l'utilisation d'Amazon S3 en tant que cible
- Limitations de l'utilisation d'Amazon S3 en tant que cible
- Sécurité
- Utilisation d'Apache Parquet pour stocker des objets Amazon S3
- Balisage d'objets Amazon S3
- Création de clés AWS KMS pour chiffrer des objets cibles Amazon S3
- Utilisation du partitionnement de dossiers basé sur la date
- Chargement parallèle de sources partitionnées lors de l'utilisation d'Amazon S3 en tant que cible pour AWS DMS
- Paramètres de point de terminaison lors de l'utilisation d'Amazon S3 en tant que cible pour AWS DMS
- Utilisation d'AWS Glue Data Catalog avec une cible Amazon S3 pour AWS DMS
- Utilisation du chiffrement des données, des fichiers Parquet et de la CDC sur votre cible Amazon S3
- Indication des opérations de base de données source dans des données S3 migrées
- Types de données cibles pour S3 Parquet

Prérequis pour l'utilisation d'Amazon S3 en tant que cible

Avant d'utiliser Amazon S3 en tant que cible, vérifiez que les conditions suivantes sont réunies :

- Le compartiment S3 que vous utilisez en tant que cible se trouve dans la même région AWS que l'instance de réplication DMS que vous utilisez pour la migration de vos données.
- Le compte AWS que vous utilisez pour la migration bénéficie d'un rôle IAM avec accès en lecture et en écriture sur le compartiment S3 que vous utilisez en tant que cible.
- Ce rôle dispose d'un accès de balisage afin que vous pouvez baliser tous les objets S3 écrits sur le compartiment cible.
- DMS (dms.amazonaws.com) a été ajouté au rôle IAM en tant qu'entité de confiance.

Pour définir cet accès de compte, assurez-vous que le rôle affecté au compte utilisateur utilisé pour créer la tâche de migration bénéficie de l'ensemble d'autorisations suivant.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:DeleteObject",
                "s3:PutObjectTagging"
            ],
            "Resource": [
                "arn:aws:s3:::buckettest2/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::buckettest2"
            ]
        }
    ]
}
```

Pour connaître les conditions préalables à l'utilisation de la validation avec S3 en tant que cible, consultez Conditions préalables à la validation des cibles S3.

Limitations de l'utilisation d'Amazon S3 en tant que cible

Les limitations suivantes s'appliquent lorsque vous utilisez Amazon S3 en tant que cible :

- N'activez pas la gestion des versions pour S3. Si vous avez besoin de la gestion des versions S3, utilisez des politiques de cycle de vie pour supprimer les anciennes versions de manière active. Dans le cas contraire, la connexion de test du point de terminaison risque d'échouer en raison du délai d'attente d'appel `list-object` S3. Pour créer une politique de cycle de vie pour un compartiment S3, consultez Gestion du cycle de vie de votre stockage. Pour supprimer une version

d'un objet S3, consultez Suppression des versions d'objet d'un compartiment activé pour la gestion des versions.

- Un compartiment S3 compatible VPC (VPC de passerelle) est pris en charge dans les versions 3.4.7 et ultérieures.
- Les commandes de langage de définition de données (DDL) suivantes sont prises en charge pour la capture des données de modification (CDC) : tronquer une table, supprimer une table, créer une table, renommer une table, ajouter une colonne, supprimer une colonne, renommer une colonne et modifier le type de données de colonne. Notez que lorsqu'une colonne est ajoutée, supprimée ou renommée sur la base de données source, aucune instruction ALTER n'est enregistrée dans le compartiment S3 cible et AWS DMS ne modifie pas les enregistrements créés précédemment pour qu'ils correspondent à la nouvelle structure. Après la modification, AWS DMS crée de nouveaux enregistrements à l'aide de la nouvelle structure de table.

  > **Note**
  >
  > Une opération DDL de troncation supprime tous les fichiers et les dossiers de table correspondants d'un compartiment S3. Vous pouvez utiliser les paramètres de tâche pour désactiver ce comportement et configurer la façon dont DMS gère le comportement DDL lors de la capture des données de modification (CDC). Pour de plus amples informations, veuillez consulter Paramètres de tâche pour la gestion du DDL de traitement des modifications.

- Le Mode LOB complet n'est pas pris en charge.
- Les modifications apportées à la structure de la table source lors du chargement complet ne sont pas prises en charge. Les modifications apportées aux données sont prises en charge pendant le chargement complet.
- Lorsque plusieurs tâches répliquent des données d'une même table source vers le même compartiment de point de terminaison S3 cible, ces tâches écrivent dans le même fichier. Nous vous recommandons de spécifier des points de terminaison (compartiments) cibles différents si votre source de données est de la même table.
- `BatchApply` n'est pas pris en charge pour un point de terminaison S3. L'utilisation de l'application par lots (par exemple, le paramètre de tâche de métadonnées cible `BatchApplyEnabled`) pour une cible S3 peut entraîner une perte de données.
- Vous ne pouvez pas utiliser `DatePartitionEnabled` ou `addColumnName` ensemble avec `PreserveTransactions` ou `CdcPath`.

- AWS DMS ne permet pas de renommer plusieurs tables sources dans le même dossier cible à l'aide de règles de transformation.
- En cas d'écriture intensive dans la table source pendant la phase de chargement complet, DMS peut écrire des enregistrements dupliqués dans le compartiment S3 ou des modifications mises en cache.
- Si vous configurez la tâche avec `TargetTablePrepMode` défini sur `DO_NOTHING`, DMS peut écrire des enregistrements dupliqués dans le compartiment S3 si la tâche s'arrête et reprend brusquement pendant la phase de chargement complet.
- Si vous configurez le point de terminaison cible avec le paramètre `PreserveTransactions` défini sur `true`, le rechargement d'une table n'efface pas les fichiers CDC générés précédemment. Pour de plus amples informations, veuillez consulter Capture des données de modification (CDC) avec l'ordre des transactions sur la cible S3.

Pour connaître les limitations de l'utilisation de la validation avec S3 en tant que cible, consultez Limitations liées à l'utilisation de la validation des cibles S3.

Sécurité

Pour utiliser Amazon S3 en tant que cible, le compte utilisé pour la migration doit bénéficier d'un accès pour l'écriture et la suppression au compartiment Amazon S3 qui est utilisé comme cible. Spécifiez l'Amazon Resource Name (ARN) du rôle IAM qui bénéficie des autorisations requises pour accéder à Amazon S3.

AWS DMS prend en charge un ensemble d'octrois prédéfinis pour Amazon S3, appelé liste de contrôle d'accès (ACL). Chaque liste ACL prédéfinie possède un ensemble de bénéficiaires et d'autorisations que vous pouvez utiliser pour définir les autorisations du compartiment Amazon S3. Vous pouvez spécifier une liste ACL prédéfinie à l'aide de l'attribut `cannedAclForObjects` de la chaîne de connexion de votre point de terminaison cible S3. Pour plus d'informations sur l'utilisation de l'attribut de connexion supplémentaire `cannedAclForObjects`, consultez Paramètres de point de terminaison lors de l'utilisation d'Amazon S3 en tant que cible pour AWS DMS. Pour plus d'informations sur les listes ACL prédéfinies Amazon S3, consultez Listes ACL prédéfinies.

Le rôle IAM que vous utilisez pour la migration doit être en mesure d'effectuer l'opération d'API `s3:PutObjectAcl`.

Utilisation d'Apache Parquet pour stocker des objets Amazon S3

Le format .csv (valeurs séparées par des virgules) est le format de stockage par défaut pour les objets cibles Amazon S3. Pour un stockage plus compact et des requêtes plus rapides, vous pouvez utiliser le format de stockage Apache Parquet (.parquet).

Apache Parquet est un format de stockage de fichiers open source, initialement conçu pour Hadoop. Pour plus d'informations sur Apache Parquet, consultez https://parquet.apache.org/.

Pour définir .parquet comme format de stockage pour vos objets cibles S3 migrés, vous pouvez utiliser les mécanismes suivants :

- Paramètres de point de terminaison que vous fournissez sous forme de paramètres d'objet JSON lorsque vous créez le point de terminaison à l'aide de l'AWS CLI ou de l'API pour AWS DMS. Pour de plus amples informations, veuillez consulter Utilisation du chiffrement des données, des fichiers Parquet et de la CDC sur votre cible Amazon S3.
- Attributs de connexion supplémentaires que vous fournissez sous forme de liste séparée par des points-virgules lorsque vous créez le point de terminaison. Pour de plus amples informations, veuillez consulter Paramètres de point de terminaison lors de l'utilisation d'Amazon S3 en tant que cible pour AWS DMS.

Balisage d'objets Amazon S3

Vous pouvez baliser des objets Amazon S3 créés par une instance de réplication en spécifiant les objets JSON appropriés dans le cadre des règles de mappage des tâches aux tables. Pour plus d'informations sur les exigences et les options du balisage d'objets S3, y compris les noms de balise valides, consultez Balisage d'objets dans le Guide de l'utilisateur Amazon Simple Storage Service. Pour plus d'informations sur le mappage de table avec JSON, consultez Spécification des règles de sélection de table et de transformation à l'aide de JSON.

Vous pouvez baliser des objets S3 créés pour des tables et des schémas spécifiés à l'aide d'un ou plusieurs objets JSON du type de règle `selection`. Vous pouvez faire suivre cet objet (ou ces objets) `selection` d'un ou plusieurs objets JSON du type de règle `post-processing` avec l'action `add-tag`. Ces règles de post-traitement identifient les objets S3 que vous souhaitez baliser et spécifient les noms et les valeurs des balises que vous souhaitez ajouter à ces objets S3.

Vous trouverez les paramètres à spécifier dans les objets JSON du type de règle `post-processing` dans le tableau suivant.

Paramètre	Valeurs possibles	Description
`rule-type`	`post-processing`	Valeur qui applique des actions de post-traitement aux objets cibles générés. Vous pouvez spécifier une ou plusieurs règles de post-traitement pour baliser les objets S3 sélectionnés.
`rule-id`	Valeur numérique.	Valeur numérique unique pour identifier la règle.
`rule-name`	Valeur alphanumérique.	Nom unique pour identifier la règle.
`rule-action`	`add-tag`	Action de post-traitement que vous souhaitez appliquer à l'objet S3. Vous pouvez ajouter une ou plusieurs balises à l'aide d'un seul objet de post-traitement JSON pour l'action `add-tag`.
`object-locator`	`schema-name` : nom du schéma de table. `table-name` : nom de la table.	Nom de chaque schéma et de chaque table auxquels la règle s'applique. Vous pouvez utiliser le signe « % » comme caractère générique pour tout ou partie de la valeur de chaque paramètre `object-locator` . Ainsi, vous pouvez faire correspondre ces éléments : • Une table unique dans un schéma unique • Une table unique dans certains schémas ou dans tous les schémas

Paramètre	Valeurs possibles	Description
		• Certaines tables ou toutes les tables dans un schéma unique • Certaines tables ou toutes les tables dans certains schémas ou dans tous les schémas

Paramètre	Valeurs possibles	Description
`tag-set`	`key` : tout nom valide pour une balise unique. `value` : toute valeur JSON valide pour cette balise.	Noms et valeurs d'un ou plusieurs balises que vous souhaitez définir sur chaque objet S3 créé qui correspond au paramètre `object-locator` spécifié. Vous pouvez spécifier jusqu'à 10 paires clé-valeur dans un objet de paramètre `tag-set` unique. Pour plus d'informations sur le balisage d'objets S3, consultez Balisage d'objets dans le Guide de l'utilisateur Amazon Simple Storage Service. Vous pouvez également spécifier une valeur dynamique pour tout ou partie de la valeur pour les paramètres `value` et `key` d'une balise à l'aide de `${dyn-value }`. Ici, `${dyn-value }` peut être `${schema-name}` ou `${table-name}` . Par conséquent, vous pouvez insérer le nom du schéma ou de la table actuellement sélectionné en tant que tout ou partie de la valeur du paramètre. **Note** **Important** Si vous insérez une valeur dynamique pour le paramètre `key`, vous pouvez générer des balises avec des noms

Paramètre	Valeurs possibles	Description
		dupliqués pour un objet S3, en fonction de la manière dont vous l'utilisez. Dans ce cas, seul un des paramètres de balise dupliqués est ajouté à l'objet.

Lorsque vous spécifiez plusieurs types de règle `post-processing` pour baliser une sélection d'objets S3, chaque objet S3 est balisé à l'aide d'un seul objet `tag-set` à partir d'une règle de post-traitement. L'ensemble de balises utilisé pour baliser un objet S3 donné est celui issu de la règle de post-traitement dont le localisateur d'objet associé correspond le mieux à l'objet S3.

Par exemple, supposons que deux règles de post-traitement identifient le même objet S3. Supposons également que le localisateur d'objet d'une règle utilise des caractères génériques et que le localisateur d'objet de l'autre règle utilise une correspondance exacte pour identifier l'objet S3 (sans caractères génériques). Dans ce cas, l'ensemble de balises associé à la règle de post-traitement avec la correspondance exacte est utilisé pour baliser l'objet S3. Si plusieurs règles de post-traitement correspondent de manière identique à un objet S3, l'ensemble de balises associé à la première règle de post-traitement est utilisé pour baliser l'objet.

Example Ajout de balises statiques à un objet S3 créé pour une table et un schéma unique

Les règles de sélection et de post-traitement suivantes ajoutent trois balises (`tag_1`, `tag_2` et `tag_3` avec leurs valeurs statiques correspondantes `value_1`, `value_2` et `value_3`) à un objet S3 créé. Cet objet S3 correspond à une table unique dans la source nommée `STOCK` avec un schéma nommé `aat2`.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "5",
            "rule-name": "5",
```

```
            "object-locator": {
                "schema-name": "aat2",
                "table-name": "STOCK"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "post-processing",
            "rule-id": "41",
            "rule-name": "41",
            "rule-action": "add-tag",
            "object-locator": {
                "schema-name": "aat2",
                "table-name": "STOCK"
            },
            "tag-set": [
              {
                "key": "tag_1",
                "value": "value_1"
              },
              {
                "key": "tag_2",
                "value": "value_2"
              },
              {
                "key": "tag_3",
                "value": "value_3"
              }
           ]
        }
    ]
}
```

Example Ajout de balises statiques et dynamiques à des objets S3 créés pour plusieurs tables et schémas

L'exemple suivant présente une règle de sélection et deux règles de post-traitement, où l'entrée de la source inclut toutes les tables et tous leurs schémas.

```
{
    "rules": [
        {
            "rule-type": "selection",
```

```
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "post-processing",
            "rule-id": "21",
            "rule-name": "21",
            "rule-action": "add-tag",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%",
            },
            "tag-set": [
              {
                "key": "dw-schema-name",
                "value":"${schema-name}"
              },
              {
                "key": "dw-schema-table",
                "value": "my_prefix_${table-name}"
              }
            ]
        },
        {
            "rule-type": "post-processing",
            "rule-id": "41",
            "rule-name": "41",
            "rule-action": "add-tag",
            "object-locator": {
                "schema-name": "aat",
                "table-name": "ITEM",
            },
            "tag-set": [
              {
                "key": "tag_1",
                "value": "value_1"
              },
              {
                "key": "tag_2",
```

```
                "value": "value_2"
              }           ]
        }
    ]
}
```

La première règle de post-traitement ajoute deux balises (`dw-schema-name` et `dw-schema-table`) avec les valeurs dynamiques correspondantes (`${schema-name}` et `my_prefix_${table-name}`) pour presque tous les objets S3 créés dans la cible. L'exception est l'objet S3 identifié et balisé avec la seconde règle de post-traitement. Par conséquent, chaque objet S3 cible identifié par le localisateur d'objet utilisant des caractères génériques est créé avec des balises qui identifient le schéma et la table auxquels il correspond dans la source.

La deuxième règle de post-traitement ajoute `tag_1` et `tag_2` avec les valeurs statiques correspondantes `value_1` et `value_2` à un objet S3 qui est identifié par un localisateur d'objet utilisant une correspondance exacte. Cet objet S3 créé correspond donc à la table unique dans la source nommée `ITEM` avec un schéma nommé `aat`. En raison de la correspondance exacte, ces balises remplacent toutes les balises sur cet objet ajouté à partir de la première règle de post-traitement, qui correspond à des objets S3 par caractère générique uniquement.

Example Ajout de noms et de valeurs de balise dynamique à des objets S3

L'exemple suivant comporte deux règles de sélection et une règle de post-traitement. Ici, l'entrée à partir de la source comprend seulement la table `ITEM` dans le schéma `retail` ou `wholesale`.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "retail",
                "table-name": "ITEM"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
```

```
                "schema-name": "wholesale",
                "table-name": "ITEM"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "post-processing",
            "rule-id": "21",
            "rule-name": "21",
            "rule-action": "add-tag",
            "object-locator": {
                "schema-name": "%",
                "table-name": "ITEM",
            },
            "tag-set": [
              {
                "key": "dw-schema-name",
                "value":"${schema-name}"
              },
              {
                "key": "dw-schema-table",
                "value": "my_prefix_ITEM"
              },
              {
                "key": "${schema-name}_ITEM_tag_1",
                "value": "value_1"
              },
              {
                "key": "${schema-name}_ITEM_tag_2",
                "value": "value_2"
              }
            ]
    ]
}
```

L'ensemble de balises pour la règle de post-traitement ajoute deux balises (`dw-schema-name` et `dw-schema-table`) à tous les objets S3 créés pour la table `ITEM` dans la cible. La première balise a la valeur dynamique `"${schema-name}"` et la deuxième balise a une valeur statique, `"my_prefix_ITEM"`. Par conséquent, chaque objet S3 cible est créé avec des balises qui identifient le schéma et la table auxquels il correspond dans la source.

En outre, l'ensemble de balises ajoute deux balises supplémentaires avec des noms dynamiques (`${schema-name}_ITEM_tag_1` et `"${schema-name}_ITEM_tag_2"`). Ces derniers ont les

valeurs statiques correspondantes `value_1` et `value_2`. Ainsi, chacune de ces balises est nommée pour le schéma actuel, `retail` ou `wholesale`. Vous ne pouvez pas créer un nom de balise dynamique dupliqué dans cet objet, car chaque objet est créé pour un nom de schéma unique. Le nom de schéma est utilisé pour créer un autre nom de balise unique.

Création de clés AWS KMS pour chiffrer des objets cibles Amazon S3

Vous pouvez créer et utiliser des clés AWS KMS personnalisées pour chiffrer vos objets cibles Amazon S3. Après avoir créé une clé KMS, vous pouvez l'utiliser pour chiffrer des objets à l'aide de l'une des approches suivantes lorsque vous créez le point de terminaison cible S3 :

- Utilisez les options suivantes pour les objets cibles S3 (avec le format de stockage de fichiers .csv par défaut) lorsque vous exécutez la commande `create-endpoint` à l'aide de l'AWS CLI.

  ```
  --s3-settings '{"ServiceAccessRoleArn": "your-service-access-ARN",
  "CsvRowDelimiter": "\n", "CsvDelimiter": ",", "BucketFolder": "your-bucket-folder",
  "BucketName": "your-bucket-name", "EncryptionMode": "SSE_KMS",
  "ServerSideEncryptionKmsKeyId": "your-KMS-key-ARN"}'
  ```

  Ici, *`your-KMS-key-ARN`* est l'Amazon Resource Name (ARN) pour votre clé KMS. Pour de plus amples informations, veuillez consulter Utilisation du chiffrement des données, des fichiers Parquet et de la CDC sur votre cible Amazon S3.
- Définissez l'attribut de connexion supplémentaire `encryptionMode` à la valeur `SSE_KMS` et l'attribut de connexion supplémentaire `serverSideEncryptionKmsKeyId` à l'ARN de votre clé KMS. Pour de plus amples informations, veuillez consulter Paramètres de point de terminaison lors de l'utilisation d'Amazon S3 en tant que cible pour AWS DMS.

Pour chiffrer des objets cibles Amazon S3 à l'aide d'une clé KMS, vous avez besoin d'un rôle IAM qui dispose des autorisations d'accès au compartiment Amazon S3. Ce rôle IAM est ensuite accessible dans une stratégie (stratégie de clé) attachée à la clé de chiffrement que vous créez. Pour ce faire, créez les éléments suivants dans votre console IAM :

- Une stratégie avec les autorisations d'accès au compartiment Amazon S3.
- Un rôle IAM avec cette stratégie.
- Une clé de chiffrement de clé KMS avec une stratégie de clé qui fait référence à ce rôle.

Les procédures suivantes décrivent la marche à suivre.

Pour créer une politique IAM avec les autorisations d'accès au compartiment Amazon S3

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation, choisissez Stratégies. La page Stratégies s'ouvre.
3. Choisissez Créer une politique. La page Créer une politique s'ouvre.
4. Choisissez Service, puis S3. Une liste d'autorisations d'action s'affiche.
5. Choisissez Développer tout pour développer la liste et choisissez au moins les autorisations suivantes :

   - ListBucket
   - PutObject
   - DeleteObject

   Choisissez toute autre autorisation dont vous avez besoin, puis sélectionnez Réduire tout pour réduire la liste.
6. Choisissez Ressources pour spécifier les ressources auxquelles vous souhaitez accéder. Choisissez au moins Toutes les ressources pour fournir un accès général aux ressources Amazon S3.
7. Ajoutez toute autre condition ou autorisation dont vous avez besoin, puis choisissez Examiner une stratégie. Vérifiez vos résultats sur la page Examiner une stratégie.
8. Si les paramètres vous conviennent, entrez un nom pour la stratégie (par exemple, `DMS-S3-endpoint-access`) et une description, puis choisissez Créer une stratégie. La page Stratégies s'ouvre et affiche un message indiquant que votre stratégie a été créée.
9. Recherchez et choisissez le nom de la stratégie dans la liste Stratégies. La page Récapitulatif affiche la stratégie au format JSON comme ci-après.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:ListBucket",
                "s3:DeleteObject"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

Vous avez créé une nouvelle stratégie pour accéder à des ressources Amazon S3 en vue du chiffrement avec un nom spécifié, par exemple `DMS-S3-endpoint-access`.

Pour créer un rôle IAM avec cette stratégie

1. Dans votre console IAM, choisissez Rôles dans le volet de navigation. La page détaillée Rôles s'ouvre.
2. Sélectionnez Créer un rôle. La page Créer un rôle s'ouvre.
3. Sélectionnez le service AWS en tant qu'entité de confiance, puis choisissez DMS comme le service qui utilisera le rôle IAM.
4. Sélectionnez Next: Permissions (Étape suivante : autorisations). La vue Attacher des stratégies d'autorisation s'affiche dans la page Créer un rôle.
5. Recherchez et sélectionnez la stratégie IAM pour le rôle IAM que vous avez créé dans la procédure précédente (`DMS-S3-endpoint-access`).
6. Choisissez Suivant : Balises. La vue Ajouter des balises s'affiche dans la page Créer un rôle. Ici, vous pouvez ajouter toutes les balises souhaitées.
7. Choisissez Suivant : vérification. La vue Vérifier s'affiche dans la page Créer un rôle. Ici, vous pouvez vérifier les résultats.
8. Si les paramètres vous conviennent, entrez un nom pour le rôle (par exemple, `DMS-S3-endpoint-access-role`) et une description, puis choisissez Créer un rôle. La page détaillée Rôles s'ouvre et affiche un message indiquant que votre rôle a été créé.

Vous avez désormais créé le nouveau rôle pour accéder aux ressources Amazon S3 en vue du chiffrement avec un nom spécifié, par exemple `DMS-S3-endpoint-access-role`.

Pour créer une clé de chiffrement de clé KMS avec une stratégie de clé qui fait référence à votre rôle IAM

> **Note**
>
> Pour plus d'informations sur le fonctionnement d'AWS DMS avec les clés de chiffrement AWS KMS, consultez Définition d'une clé de chiffrement et spécification des autorisations AWS KMS.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Choisissez Create key. La page Configurer la clé s'ouvre.
5. Pour Type de clé, choisissez Symétrique.

   > **Note**
   >
   > Vous ne pouvez créer qu'une clé symétrique, car tous les services AWS, tels qu'Amazon S3, ne fonctionnent qu'avec des clés de chiffrement symétriques.

6. Choisissez Options avancées. Pour Origine des éléments de clé, assurez-vous que KMS est choisi, puis choisissez Suivant. La page Ajouter des étiquettes s'ouvre.
7. Dans Créer un alias et une description, entrez un alias pour la clé (par exemple, `DMS-S3-endpoint-encryption-key`) et toute description supplémentaire.
8. Pour Balises, ajoutez les balises que vous souhaitez pour vous aider à identifier la clé et suivre son utilisation, puis choisissez Suivant. La page Définir des autorisations d'administration de clé s'ouvre et affiche une liste d'utilisateurs et de rôles parmi lesquels vous pouvez choisir.
9. Ajoutez les utilisateurs et les rôles que vous souhaitez voir gérer la clé. Assurez-vous que ces utilisateurs et ces rôles ont les autorisations requises pour gérer la clé.
10. Pour Suppression de clé, choisissez si les administrateurs de clé peuvent supprimer celle-ci, puis choisissez Suivant. La page Définir des autorisations d'utilisation de clé s'ouvre et affiche une liste supplémentaire d'utilisateurs et de rôles parmi lesquels vous pouvez choisir.

11. Pour Ce compte, choisissez les utilisateurs disponibles pour lesquels vous souhaitez effectuer des opérations de chiffrement sur des cibles Amazon S3. Choisissez également le rôle que vous avez créé précédemment dans Rôles pour activer l'accès au chiffrement des objets cibles Amazon S3, par exemple `DMS-S3-endpoint-access-role`.
12. Si vous souhaitez ajouter d'autres comptes non répertoriés pour avoir ce même accès, pour Autres comptes AWS, choisissez Ajouter un autre compte AWS, puis Suivant. La page Réviser et modifier la stratégie de clé s'ouvre, affichant le JSON de la stratégie de clé que vous pouvez réviser et modifier en saisissant les informations dans le JSON existant. Ici, vous pouvez voir où la stratégie clé fait référence au rôle et aux utilisateurs (par exemple, `Admin` et `User1`) que vous avez choisis à l'étape précédente. Vous pouvez également voir les actions clés autorisées pour les différents mandataires (utilisateurs et rôles), comme illustré dans l'exemple suivant.

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:root"
        ]
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow access for Key Administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/Admin"
        ]
      },
      "Action": [
        "kms:Create*",
        "kms:Describe*",
        "kms:Enable*",
        "kms:List*",
        "kms:Put*",
```

```
        "kms:Update*",
        "kms:Revoke*",
        "kms:Disable*",
        "kms:Get*",
        "kms:Delete*",
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ScheduleKeyDeletion",
        "kms:CancelKeyDeletion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/DMS-S3-endpoint-access-role",
          "arn:aws:iam::111122223333:role/Admin",
          "arn:aws:iam::111122223333:role/User1"
        ]
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow attachment of persistent resources",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/DMS-S3-endpoint-access-role",
          "arn:aws:iam::111122223333:role/Admin",
          "arn:aws:iam::111122223333:role/User1"
        ]
      },
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
```

```
        "kms:RevokeGrant"
      ],
      "Resource": "*",
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": true
        }
      }
    }
  ]
```

13. Choisissez Finish (Terminer). La page Clés de chiffrement s'ouvre et affiche un message indiquant que votre clé KMS a été créée.

Vous venez de créer une nouvelle clé KMS avec un alias spécifié (par exemple, `DMS-S3-endpoint-encryption-key`). Cette clé permet à AWS DMS de chiffrer des objets cibles Amazon S3.

Utilisation du partitionnement de dossiers basé sur la date

AWS DMS prend en charge les partitions de dossiers S3 en fonction d'une date de validation des transactions lorsque vous utilisez Amazon S3 en tant que point de terminaison cible. Avec le partitionnement de dossiers basé sur la date, vous pouvez écrire des données à partir d'une table source unique dans une structure de dossiers hiérarchisée dans le temps dans un compartiment S3. En partitionnant des dossiers lors de la création d'un point de terminaison cible S3, vous pouvez :

- Mieux gérer vos objets S3
- Limiter la taille de chaque dossier S3
- Optimiser les requêtes du lac de données ou les autres opérations ultérieures

Vous pouvez activer le partitionnement de dossiers basé sur la date lorsque vous créez un point de terminaison cible S3. Vous pouvez l'activer lorsque vous migrez des données existantes et que vous répliquez les modifications continues (chargement complet + CDC) ou lorsque vous répliquez uniquement les modifications de données (CDC uniquement). Utilisez les paramètres de point de terminaison cible suivants :

- `DatePartitionEnabled` : spécifie le partitionnement en fonction des dates. Définissez cette option booléenne sur `true` pour partitionner les dossiers de compartiment S3 en fonction des dates de validation des transactions.

Vous ne pouvez pas utiliser ce paramètre avec `PreserveTransactions` ou `CdcPath`.

La valeur par défaut est `false`.

- `DatePartitionSequence` : identifie la séquence du format de date à utiliser lors du partitionnement des dossiers. Définissez cette option ENUM sur `YYYYMMDD`, `YYYYMMDDHH`, `YYYYMM`, `MMYYYYDD` ou `DDMMYYYY`. La valeur par défaut est `YYYYMMDD`. Utilisez ce paramètre lorsque `DatePartitionEnabled` est défini sur `true`..
- `DatePartitionDelimiter` : spécifie un délimiteur de date à utiliser lors du partitionnement des dossiers. Définissez cette option ENUM sur `SLASH`, `DASH`, `UNDERSCORE` ou `NONE`. La valeur par défaut est `SLASH`. Utilisez ce paramètre lorsque `DatePartitionEnabled` est défini sur `true`.

L'exemple suivant montre comment activer le partitionnement de dossiers basé sur la date, avec les valeurs par défaut pour la séquence de partition de données et le délimiteur. Il utilise l'option `--s3-settings '{`*`json-settings`*`}'` de la commande AWS CLI `create-endpoint`.

```
  --s3-settings '{"DatePartitionEnabled": true,"DatePartitionSequence":
"YYYYMMDD","DatePartitionDelimiter": "SLASH"}'
```

Chargement parallèle de sources partitionnées lors de l'utilisation d'Amazon S3 en tant que cible pour AWS DMS

Vous pouvez configurer un chargement complet parallèle de sources de données partitionnées sur les cibles Amazon S3. Cette approche améliore les temps de chargement lors de la migration des données partitionnées à partir des moteurs de base de données source pris en charge vers la cible S3. Pour améliorer les temps de chargement des données sources partitionnées, vous créez des sous-dossiers cibles S3 mappés aux partitions de chaque table de la base de données source. Ces sous-dossiers liés à une partition permettent à AWS DMS d'exécuter des processus parallèles pour remplir chaque sous-dossier de la cible.

Pour configurer le chargement complet parallèle d'une cible S3, S3 prend en charge trois types de règles `parallel-load` pour la règle `table-settings` de mappage de table :

- `partitions-auto`
- `partitions-list`
- `ranges`

Pour plus d'informations sur ces types de règles de chargement parallèle, consultez Règles des paramètres de table et de collection et opérations.

Pour les types de règles `partitions-auto` et `partitions-list`, AWS DMS utilise le nom de chaque partition depuis le point de terminaison source pour identifier la structure des sous-dossiers cibles, comme suit.

```
bucket_name/bucket_folder/database_schema_name/table_name/partition_name/
LOADseq_num.csv
```

Ici, le chemin du sous-dossier où les données sont migrées et stockées sur la cible S3 inclut un sous-dossier *`partition_name`* supplémentaire qui correspond à une partition source portant le même nom. Ce sous-dossier *`partition_name`* stocke ensuite un ou plusieurs fichiers `LOAD`*`seq_num`*`.csv` contenant des données migrées depuis la partition source spécifiée. Ici, *`seq_num`* est le suffixe du numéro de séquence sur le nom du fichier .csv, comme `00000001` dans le fichier .csv portant le nom `LOAD00000001.csv`.

Cependant, certains moteurs de base de données, tels que MongoDB et DocumentDB, n'utilisent pas le concept de partitions. Pour ces moteurs de base de données, AWS DMS ajoute l'index du segment source en cours d'exécution en tant que préfixe au nom du fichier .csv cible, comme suit.

```
.../database_schema_name/table_name/SEGMENT1_LOAD00000001.csv
.../database_schema_name/table_name/SEGMENT1_LOAD00000002.csv
...
.../database_schema_name/table_name/SEGMENT2_LOAD00000009.csv
.../database_schema_name/table_name/SEGMENT3_LOAD0000000A.csv
```

Ici, les fichiers `SEGMENT1_LOAD00000001.csv` et `SEGMENT1_LOAD00000002.csv` sont nommés avec le même préfixe d'index de segment source en cours d'exécution, `SEGMENT1`. Ils portent ce nom, car les données sources migrées pour ces deux fichiers .csv sont associées au même index de segment source en cours d'exécution. D'autre part, les données migrées stockées dans chacun des fichiers cibles `SEGMENT2_LOAD00000009.csv` et `SEGMENT3_LOAD0000000A.csv` sont associées à différents index de segment source en cours d'exécution. Le nom de chaque fichier est préfixé avec le nom de son index de segment en cours d'exécution, `SEGMENT2` et `SEGMENT3`.

Pour le type de chargement parallèle `ranges`, vous définissez les noms et les valeurs de colonne à l'aide des paramètres `columns` et `boundaries` des règles `table-settings`. Avec ces règles, vous pouvez spécifier des partitions correspondant aux noms de segment, comme suit.

```
"parallel-load": {
    "type": "ranges",
    "columns": [
         "region",
         "sale"
    ],
    "boundaries": [
          [
               "NORTH",
               "1000"
          ],
          [
               "WEST",
               "3000"
          ]
    ],
    "segment-names": [
          "custom_segment1",
          "custom_segment2",
          "custom_segment3"
    ]
}
```

Ici, le paramètre `segment-names` définit les noms de trois partitions pour migrer les données en parallèle sur la cible S3. Les données migrées sont chargées en parallèle et stockées dans des fichiers .csv sous les sous-dossiers de partition, comme suit.

```
.../database_schema_name/table_name/custom_segment1/LOAD[00000001...].csv
.../database_schema_name/table_name/custom_segment2/LOAD[00000001...].csv
.../database_schema_name/table_name/custom_segment3/LOAD[00000001...].csv
```

Ici, AWS DMS stocke une série de fichiers .csv dans chacun des trois sous-dossiers de partition. La série de fichiers .csv de chaque sous-dossier de partition est nommée de manière incrémentielle à partir de `LOAD00000001.csv` jusqu'à ce que toutes les données soient migrées.

Dans certains cas, il se peut que vous ne nommiez pas explicitement les sous-dossiers de partition pour un type de chargement parallèle `ranges` à l'aide du paramètre `segment-names`. Dans ce cas, AWS DMS applique le comportement par défaut qui consiste à créer chaque série de fichiers .csv dans son sous-dossier *`table_name`*. Ici, AWS DMS préfixe les noms de fichier de chaque série de fichiers .csv avec le nom de l'index du segment source en cours d'exécution, comme suit.

```
.../database_schema_name/table_name/SEGMENT1_LOAD[00000001...].csv
.../database_schema_name/table_name/SEGMENT2_LOAD[00000001...].csv
.../database_schema_name/table_name/SEGMENT3_LOAD[00000001...].csv
...
.../database_schema_name/table_name/SEGMENTZ_LOAD[00000001...].csv
```

Paramètres de point de terminaison lors de l'utilisation d'Amazon S3 en tant que cible pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données cible Amazon S3 comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison cible à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--s3-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec Amazon S3 en tant que cible sont indiqués dans le tableau suivant.

Option	Description
`CsvNullValue`	Un paramètre facultatif qui indique comment AWS DMS traite des valeurs nulles (null). Lors de la gestion de la valeur null (nulle), vous pouvez utiliser ce paramètre pour transmettre une chaîne définie par l'utilisateur comme nulle lors de l'écriture sur la cible. Par exemple, lorsque les colonnes cibles peuvent être définies comme nulles (nullable), vous pouvez utiliser cette option pour différencier la valeur de chaîne vide de la valeur nulle. Ainsi, si vous donnez à ce paramètre la valeur d'une chaîne vide (" " ou ''), AWS DMS traite la chaîne vide comme la valeur null (nulle) au lieu de `NULL`. Valeur par défaut : `NULL` Valeurs valides : toutes les chaînes valides Exemple : `--s3-settings '{"CsvNullValue": " "}'`
`AddColumnName`	Paramètre facultatif qui lorsqu'il est défini sur `true` ou y vous permet d'ajouter les informations de nom de colonne dans le fichier de sortie .csv.

Option	Description
	Vous ne pouvez pas utiliser ce paramètre avec `PreserveTransactions` ou `CdcPath`. Valeur par défaut : `false` Valeurs valides : `true`, `false`, `y`, `n` Exemple : `--s3-settings '{"AddColumnName": true}'`
`AddTrailingPaddingCharacter`	Utilisez le paramètre de point de terminaison cible S3 `AddTrailingPaddingCharacter` pour ajouter un remplissage aux données de chaîne. La valeur par défaut est `false`. Type : booléen Exemple : `--s3-settings '{"AddTrailingPaddingCharacter": true}'`
`BucketFolder`	Un paramètre facultatif pour définir le nom d'un dossier dans le compartiment S3. S'il est fourni, les objets cibles sont créés en tant que fichiers .csv ou .parquet dans le chemin *`BucketFolder`* `/`*`schema_name`* `/`*`table_name`* `/`. Si ce paramètre n'est pas spécifié, le chemin utilisé est *`schema_name`* `/`*`table_name`* `/`. Exemple : `--s3-settings '{"BucketFolder": "testFolder"}'`
`BucketName`	Nom du compartiment S3 dans lequel les objets cibles S3 sont créés en tant que fichiers .csv ou .parquet. Exemple : `--s3-settings '{"BucketName": "buckettest"}'`

Option	Description
`CannedAcl ForObjects`	Permet à AWS DMS de spécifier une liste de contrôle d'accès prédéfini e pour les objets créés dans le compartiment S3 en tant que fichiers .csv ou .parquet. Pour plus d'informations sur les listes ACL Amazon S3 prédéfinies, veuillez consulter la rubrique Listes ACL prédéfinies dans le Guide du développeur Amazon S3. Valeur par défaut : NONE Les valeurs valides pour cet attribut sont : NONE, PRIVATE, PUBLIC_READ, PUBLIC_READ_WRITE, AUTHENTICATED_READ , AWS_EXEC_READ, BUCKET_OWNER_READ, BUCKET_OW NER_FULL_CONTROL. Exemple : `--s3-settings '{"CannedAclForObjects": "PUBLIC_READ"}'`

Option	Description
`CdcInsertsOnly`	Paramètre facultatif lors d'une capture de données de modification (CDC) pour écrire uniquement des opérations INSERT dans les fichiers de sortie .csv (valeurs séparées par des virgules) ou .parquet (stockage en colonnes). Par défaut (le paramètre `false`), le premier champ d'un enregistrement .csv ou .parquet contient la lettre I (INSERT), U (UPDATE) ou D (DELETE). Cette lettre indique si la ligne a été insérée, mise à jour ou supprimée dans la base de données source pour une charge CDC vers la cible. Si `cdcInsertsOnly` est défini sur `true` ou y, seules les commandes INSERT de la base de données source sont migrées vers le fichier .csv ou .parquet. Pour le format .csv uniquement, l'enregistrement de de ces commandes INSERT dépend de la valeur de `IncludeOpForFullLoad` . Si `IncludeOpForFullLoad` est défini sur `true`, le premier champ de chaque enregistrement CDC est défini sur I pour indiquer l'opérati on INSERT à la source. Si `IncludeOpForFullLoad` est défini sur `false`, chaque enregistrement CDC est écrit sans premier champ pour indiquer l'opération INSERT à la source. Pour plus d'informations sur le fonctionnement de ces paramètres utilisés conjointement, consultez Indication des opérations de base de données source dans des données S3 migrées. Valeur par défaut : `false` Valeurs valides : `true`, `false`, y, n Exemple : `--s3-settings '{"CdcInsertsOnly": true}'`

Option	Description
`CdcInsertsAndUpdates`	Permet une charge CDC (Change Data Capture) pour écrire des opérations INSERT et UPDATE dans des fichiers de sortie .csv ou .parquet (stockage en colonnes). Le paramètre par défaut est `false`, mais quand `cdcInsertsAndUpdates` est défini sur `true` ou `y`, les opérations INSERT et UPDATE de la base de données source sont migrées vers le fichier .csv ou .parquet. Pour le format de fichier .csv uniquement, la manière dont ces opérations INSERT et UPDATE sont enregistrées dépend de la valeur du paramètre `includeOpForFullLoad`. Si la valeur `includeOpForFullLoad` est définie sur `true`, le premier champ de chaque enregistrement CDC est défini sur `I` ou `U` pour indiquer les opérations INSERT et UPDATE à la source. Mais si `includeOpForFullLoad` est défini sur `false`, les enregistrements CDC sont écrits sans indication des opérations INSERT ou UPDATE à la source. Pour plus d'informations sur le fonctionnement de ces paramètres utilisés conjointement, consultez Indication des opérations de base de données source dans des données S3 migrées. **Note** `CdcInsertsOnly` et `cdcInsertsAndUpdates` ne peuvent pas tous les deux être définis sur true pour le même point de terminaison. Définissez `cdcInsertsOnly` ou `cdcInsertsAndUpdates` sur `true` pour le même point de terminaison, mais pas les deux. Valeur par défaut : `false` Valeurs valides : `true`, `false`, `y`, `n` Exemple : `--s3-settings '{"CdcInsertsAndUpdates": true}'`

Option	Description
CdcPath	Spécifie le chemin du dossier des fichiers CDC. Pour une source S3, cet attribut est requis si une tâche capture les données modifiées. Dans le cas contraire, il est facultatif. Si CdcPath est défini, DMS lit les fichiers CDC à partir de ce chemin et réplique les modifications des données sur le point de terminaison cible. Pour une cible S3, si vous définisse z PreserveTransactions sur true, DMS vérifie que vous avez défini ce paramètre sur un chemin de dossier de votre cible S3 où DMS peut enregistrer l'ordre des transactions pour le chargement CDC. DMS crée ce chemin de dossier CDC soit dans votre répertoire de travail cible S3, soit dans l'emplacement cible S3 spécifié par BucketFolder et BucketName . Vous ne pouvez pas utiliser ce paramètre avec DatePartitionEnabl ed ou AddColumnName . Type : chaîne Par exemple, si vous définissez CdcPath sur MyChangedData et BucketName sur MyTargetBucket , mais que vous ne spécifiez pas BucketFolder , DMS crée le chemin du dossier CDC suivant : MyTargetBucket/MyChangedData . Si vous définissez la même valeur pour CdcPath et que vous définisse z BucketName sur MyTargetBucket et BucketFolder sur MyTargetData , DMS crée le chemin du dossier CDC suivant : MyTargetBucket/MyTargetData/MyChangedData . **Note** Ce paramètre est pris en charge dans AWS DMS versions 3.4.2 et ultérieures. Lors de la capture des modifications de données dans l'ordre des transactions, DMS stocke toujours les modifications de ligne dans des fichiers .csv, quelle que soit la valeur du paramètre S3 DataFormat sur la cible. DMS n'enregistre pas les modificat

Option	Description
	ions de données dans l'ordre des transactions à l'aide de fichiers .parquet.
`CdcMaxBatchInterval`	Condition de durée maximale de l'intervalle, définie en secondes, à laquelle envoyer un fichier vers Amazon S3. Valeur par défaut : 60 secondes Lorsque `CdcMaxBatchInterval` et `CdcMinFileSize` sont spécifiés, l'écriture du fichier est déclenchée dès que l'une des conditions de paramètre est remplie.
`CdcMinFileSize`	Condition de taille minimale de fichier, définie en kilo-octets, à laquelle envoyer un fichier vers Amazon S3. Valeur par défaut : 32 000 Ko Lorsque `CdcMinFileSize` et `CdcMaxBatchInterval` sont spécifiés, l'écriture du fichier est déclenchée dès que l'une des conditions de paramètre est remplie.

Option	Description
`PreserveTransactions`	Lorsque la valeur est `true`, DMS enregistre l'ordre des transactions pour une capture des données de modification (CDC) sur la cible Amazon S3 spécifiée par `CdcPath`. Vous ne pouvez pas utiliser ce paramètre avec `DatePartitionEnabled` ou `AddColumnName` . Type : booléen Lors de la capture des modifications de données dans l'ordre des transactions, DMS stocke toujours les modifications de ligne dans des fichiers .csv, quelle que soit la valeur du paramètre S3 DataFormat sur la cible. DMS n'enregistre pas les modifications de données dans l'ordre des transactions à l'aide de fichiers .parquet. **Note** Ce paramètre est pris en charge dans AWS DMS versions 3.4.2 et ultérieures.

Option	Description
`IncludeOpForFullLoad`	Paramètre facultatif lors d'un chargement complet pour écrire les opérations INSERT dans les fichiers de sortie .csv (valeurs séparées par des virgules) uniquement. Pour le chargement complet, les enregistrements peuvent être insérés uniquement. Par défaut (paramètre `false`), avec un chargement complet, aucune information n'est enregistrée dans ces fichiers de sortie pour indiquer que les lignes ont été ajoutées à la base de données source. Si `IncludeOpForFullLoad` est défini sur `true` ou `y`, l'opération INSERT est enregistrée en tant que annotation I dans le premier champ du fichier .csv. **Note** Ce paramètre fonctionne conjointement avec `CdcInsertsOnly` ou `CdcInsertsAndUpdates` pour une sortie dans des fichiers .csv uniquement. Pour plus d'informations sur le fonctionnement de ces paramètres utilisés conjointement, consultez Indication des opérations de base de données source dans des données S3 migrées. Valeur par défaut : `false` Valeurs valides : `true`, `false`, `y`, `n` Exemple : `--s3-settings '{"IncludeOpForFullLoad": true}'`

Option	Description
`CompressionType`	Paramètre facultatif qui lorsqu'il est défini sur `GZIP` utilise GZIP pour compresser les fichiers cibles .csv ou .parquet. Lorsque ce paramètre est défini sur la valeur par défaut, les fichiers demeurent non compressés. Valeur par défaut : `NONE` Valeurs valides : `GZIP` ou `NONE` Exemple : `--s3-settings '{"CompressionType": "GZIP"}'`
`CsvDelimiter`	Délimiteur utilisé pour séparer les colonnes des fichiers sources .csv. La valeur par défaut est une virgule (,). Exemple : `--s3-settings '{"CsvDelimiter": ","}'`
`CsvRowDelimiter`	Délimiteur utilisé pour séparer les lignes des fichiers sources .csv. La valeur par défaut est un caractère de saut de ligne (\ n). Exemple : `--s3-settings '{"CsvRowDelimiter": "\n"}'`
`MaxFileSize`	Une valeur qui spécifie la taille maximale (en Ko) des fichiers .csv à créer lors de la migration vers une cible S3 pendant le chargement complet. Valeur par défaut : 1 048 576 Ko (1 Go) Valeurs valides : 1 à 1 048 576 Exemple : `--s3-settings '{"MaxFileSize": 512}'`

Option	Description
`Rfc4180`	Paramètre facultatif utilisé pour définir le comportement afin de se conformer à RFC pour les données migrées vers Amazon S3 à l'aide du format de fichier .csv uniquement. Lorsque ce paramètre est défini sur `true` ou `y` à l'aide d'Amazon S3 en tant que cible, si les données contiennent des guillemets, des virgules ou des caractères de saut de ligne, AWS DMS place l'ensemble de la colonne entre guillemets doubles (''). Chaque point d'interrogation contenu dans les données est répété deux fois. Ce format est conforme à RFC 4180. Valeur par défaut : `true` Valeurs valides : `true`, `false`, `y`, `n` Exemple : `--s3-settings '{"Rfc4180": false}'`
`EncryptionMode`	Mode de chiffrement côté serveur que vous souhaitez utiliser pour chiffrer vos fichiers d'objets .csv ou .parquet copiés dans S3. Les valeurs valides sont `SSE_S3` (chiffrement côté serveur S3) ou `SSE_KMS` (chiffrement de clé KMS). Si vous choisissez `SSE_KMS`, définissez le paramètre `ServerSideEncryptionKmsKeyId` sur l'Amazon Resource Name (ARN) de la clé KMS à utiliser pour le chiffrement. **Note** Vous pouvez également utiliser la commande CLI `modify-endpoint` pour modifier la valeur de l'attribut `EncryptionMode` d'un point de terminaison existant de `SSE_KMS` à `SSE_S3`. Vous ne pouvez pas modifier la valeur `EncryptionMode` de `SSE_S3` à `SSE_KMS`. Valeur par défaut : `SSE_S3` Valeurs valides : `SSE_S3` ou `SSE_KMS` Exemple : `--s3-settings '{"EncryptionMode": SSE_S3}'`

Option	Description
`ServerSideEncryptionKmsKeyId`	Si vous définissez `EncryptionMode` sur `SSE_KMS`, définissez ce paramètre sur l'Amazon Resource Name (ARN) de la clé KMS. Vous pouvez rechercher cet ARN en sélectionnant l'alias de clé dans la liste des clés AWS KMS créées pour votre compte. Lorsque vous créez la clé, vous devez associer les stratégies et les rôles spécifiques associés à cette clé KMS. Pour de plus amples informations, veuillez consulter Création de clés AWS KMS pour chiffrer des objets cibles Amazon S3. Exemple : `--s3-settings '{"ServerSideEncryptionKmsKeyId":"arn:aws:kms:us-east-1:111122223333:key/11a1a1a1-aaaa-9999-abab-2bbbbbb222a2"}'`
`DataFormat`	Format de sortie pour les fichiers utilisés par AWS DMS pour créer des objets S3. Pour les cibles Amazon S3, AWS DMS prend en charge les fichiers .csv ou .parquet. Les fichiers .parquet ont un format de stockage par colonnes binaire avec des options de compression efficaces et des performances de requête plus rapides. Pour plus d'informations sur les fichiers .parquet, consultez https://parquet.apache.org/. Valeur par défaut : `csv` Valeurs valides : `csv` ou `parquet` Exemple : `--s3-settings '{"DataFormat": "parquet"}'`

Option	Description
`EncodingType`	Type d'encodage Parquet. Parmi les options de type d'encodage, on trouve : • `rle-dictionary` : cet encodage par dictionnaire utilise une combinaison d'encodage par compression de bits et longueur d'exécuti on pour stocker de manière plus efficace les valeurs répétitives. • `plain` : pas d'encodage. • `plain-dictionary` : cet encodage par dictionnaire crée un dictionnaire à partir des valeurs rencontrées dans une colonne donnée. Le dictionnaire est stocké dans une page de dictionnaire pour chaque bloc de colonne. Valeur par défaut : `rle-dictionary` Valeurs valides : `rle-dictionary` , `plain` ou `plain-dictionary` Exemple : `--s3-settings '{"EncodingType": "plain-di ctionary"}'`
`DictPageS izeLimit`	Taille maximale autorisée, en octets, pour une page de dictionnaire dans un fichier .parquet. Si une page de dictionnaire dépasse cette valeur, elle utilise l'encodage brut. Valeur par défaut : 1 024 000 (1 Mo) Valeurs valides : Toute valeur entière valide Exemple : `--s3-settings '{"DictPageSizeLimit": 2,048,000}'`
`RowGroupLength`	Nombre de lignes dans un groupe de lignes d'un fichier .parquet. Valeur par défaut : 10 024 (10 Ko) Valeurs valides : tout entier valide Exemple : `--s3-settings '{"RowGroupLength": 20,048}'`

Option	Description
`DataPageSize`	Taille maximale autorisée, en octets, pour une page de données dans un fichier .parquet. Valeur par défaut : 1 024 000 (1 Mo) Valeurs valides : tout entier valide Exemple : `--s3-settings '{"DataPageSize": 2,048,000}'`
`ParquetVersion`	Version du format de fichier .parquet. Valeur par défaut : `PARQUET_1_0` Valeurs valides : `PARQUET_1_0` ou `PARQUET_2_0` Exemple : `--s3-settings '{"ParquetVersion": "PARQUET_2_0"}'`
`EnableStatistics`	Définissez ce paramètre sur `true` ou `y` pour activer les statistiques sur les pages et les groupes de lignes des fichiers .parquet. Valeur par défaut : `true` Valeurs valides : `true`, `false`, `y`, `n` Exemple : `--s3-settings '{"EnableStatistics": false}'`

Option	Description
`TimestampColumnName`	Paramètre facultatif permettant d'inclure une colonne d'horodatage dans les données de point de terminaison cible S3. AWS DMS inclut une colonne `STRING` supplémentaire dans les fichiers d'objet .csv ou .parquet de vos données migrées lorsque vous définissez une valeur non vide pour `TimestampColumnName` . Pour un chargement complet, chaque ligne de cette colonne d'horodat age contient une valeur d'horodatage correspondant au moment où les données ont été transférées de la source vers la cible par DMS. Pour un chargement CDC, chaque ligne de la colonne d'horodatage contient la valeur d'horodatage correspondant au moment de la validatio n de cette ligne dans la base de données source. Le format chaîne pour la valeur de cette colonne d'horodatage est `yyyy-MM-dd HH:mm:ss.SSSSSS` . Par défaut, la précision de cette valeur est en microsecondes. Pour un chargement de CDC, l'arrondi de la précision dépend de l'horodatage de validation pris en charge par DMS pour la base de données source. Lorsque le paramètre `AddColumnName` est défini sur `true`, DMS inclut également le nom de la colonne d'horodatage que vous définissez en tant que valeur non vide pour `TimestampColumnName` . Exemple : `--s3-settings '{"TimestampColumnName": "TIMESTAMP"}'`

Option	Description
`UseTaskStartTimeForFullLoadTimestamp`	Lorsqu'il est défini sur `true`, ce paramètre utilise l'heure de début de la tâche comme valeur de colonne d'horodatage au lieu de l'heure à laquelle les données sont écrites dans la cible. Pour un chargement complet, lorsque `UseTaskStartTimeForFullLoadTimestamp` est défini sur `true`, chaque ligne de la colonne d'horodatage contient l'heure de début de la tâche. Pour les chargements CDC, chaque ligne de la colonne d'horodatage contient l'heure de validation de la transaction. Lorsque `UseTaskStartTimeForFullLoadTimestamp` est défini sur `false`, l'horodatage de chargement complet dans la colonne d'horodatage augmente avec l'heure à laquelle les données arrivent sur la cible. Valeur par défaut : `false` Valeurs valides : `true`, `false` Exemple : `--s3-settings '{"UseTaskStartTimeForFullLoadTimestamp": true}'` `UseTaskStartTimeForFullLoadTimestamp: true` fait en sorte que la colonne `TimestampColumnName` de la cible S3 pour un chargement complet puisse être triée avec `TimestampColumnName` pour un chargement CDC.

Option	Description
`ParquetTimestampInMillisecond`	Paramètre facultatif qui spécifie la précision des valeurs de colonne `TIMESTAMP` écrites dans un fichier d'objet S3 au format .parquet. Lorsque cet attribut est défini sur `true` ou `y`, AWS DMS écrit toutes les colonnes `TIMESTAMP` dans un fichier au format .parquet avec une précision à la milliseconde. Sinon, DMS les écrit avec une précision à la microseconde. Actuellement, Amazon Athena et AWS Glue peuvent gérer une précision à la milliseconde pour les valeurs `TIMESTAMP` . Définissez cet attribut sur true pour les fichiers d'objet de point de terminaison S3 au format .parquet uniquement si vous prévoyez d'interroger ou de traiter les données avec Athena ou AWS Glue. **Note** • AWS DMS écrit les valeurs de colonne `TIMESTAMP` dans un fichier S3 au format .csv avec une précision à la microseconde. • La valeur de cet attribut n'a aucun effet sur le format chaîne de la valeur de colonne d'horodatage insérée en définissant l'attribut `TimestampColumnName` . Valeur par défaut : `false` Valeurs valides : `true`, `false`, `y`, `n` Exemple : `--s3-settings '{"ParquetTimestampInMillisecond": true}'`

Option	Description
`GlueCatalogGeneration`	Pour générer un catalogue AWS Glue Data Catalog, définissez ce paramètre de point de terminaison sur `true`. Valeur par défaut : `false` Valeurs valides : `true`, `false` Exemple : `--s3-settings '{"GlueCatalogGeneration": true}'` Remarque : n'utilisez pas `GlueCatalogGeneration` avec `PreserveTransactions` et `CdcPath`.

Utilisation d'AWS Glue Data Catalog avec une cible Amazon S3 pour AWS DMS

AWS Glue est un service qui fournit des méthodes simples pour classer les données et consiste en un référentiel de métadonnées appelé AWS Glue Data Catalog. Vous pouvez intégrer un catalogue AWS Glue Data Catalog au point de terminaison cible Amazon S3 et interroger les données Amazon S3 via d'autres services AWS tels qu'Amazon Athena. Amazon Redshift fonctionne avec AWS Glue, mais AWS DMS ne le prend pas en charge en tant qu'option préconçue.

Pour générer le catalogue de données, définissez le paramètre de point de terminaison `GlueCatalogGeneration` sur `true`, comme illustré dans l'exemple AWS CLI suivant.

```
aws dms create-endpoint --endpoint-identifier s3-target-endpoint
            --engine-name s3 --endpoint-type target--s3-settings
 '{"ServiceAccessRoleArn":
            "your-service-access-ARN", "BucketFolder": "your-bucket-folder",
 "BucketName":
            "your-bucket-name", "DataFormat": "parquet", "GlueCatalogGeneration":
 true}'
```

Pour une tâche de réplication à chargement complet incluant le type de données `csv`, définissez `IncludeOpForFullLoad` sur `true`.

N'utilisez pas `GlueCatalogGeneration` avec `PreserveTransactions` et `CdcPath`. Le Crawler AWS Glue ne parvient pas à rapprocher les différents schémas des fichiers stockés dans le chemin `CdcPath` spécifié.

Pour qu'Amazon Athena puisse indexer vos données Amazon S3 et pour que vous puissiez interroger vos données à l'aide de requêtes SQL standard via Amazon Athena, le rôle IAM attaché au point de terminaison doit posséder la politique suivante :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
            "Resource": [
                "arn:aws:s3:::bucket123",
                "arn:aws:s3:::bucket123/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateDatabase",
                "glue:GetDatabase",
                "glue:CreateTable",
                "glue:DeleteTable",
                "glue:UpdateTable",
                "glue:GetTable",
                "glue:BatchCreatePartition",
                "glue:CreatePartition",
                "glue:UpdatePartition",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:BatchGetPartition"
            ],
            "Resource": [
```

```
                "arn:aws:glue:*:111122223333:catalog",
                "arn:aws:glue:*:111122223333:database/*",
                "arn:aws:glue:*:111122223333:table/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "athena:StartQueryExecution",
                "athena:GetQueryExecution",
                "athena:CreateWorkGroup"
            ],
            "Resource": "arn:aws:athena:*:111122223333:workgroup/
glue_catalog_generation_for_task_*"
        }
    ]
}
```

Références

- Pour plus d'informations sur la configuration d'une règle dans AWS Glue, consultez les Concepts dans le Guide du développeur AWS Glue.
- Pour plus d'informations sur AWS Glue Data Catalog, consultez Composants dans le Manuel du développeur AWS Glue.

Utilisation du chiffrement des données, des fichiers Parquet et de la CDC sur votre cible Amazon S3

Vous pouvez utiliser les paramètres de point de terminaison cible S3 pour configurer les éléments suivants :

- Une clé KMS personnalisée pour chiffrer vos objets cibles S3.
- Des fichiers Parquet comme format de stockage pour les objets cibles S3.
- La capture des données de modification (CDC) avec l'ordre des transactions sur la cible S3.
- Intégrez un catalogue AWS Glue Data Catalog au point de terminaison cible Amazon S3 et interrogez les données Amazon S3 via d'autres services tels qu'Amazon Athena.

Des paramètres de clé AWS KMS pour le chiffrement des données

Les exemples suivants illustrent la configuration d'une clé KMS personnalisée pour chiffrer vos objets cibles S3. Pour commencer, vous pouvez exécuter la commande CLI `create-endpoint` suivante.

```
aws dms create-endpoint --endpoint-identifier s3-target-endpoint --engine-name s3 --
endpoint-type target
--s3-settings '{"ServiceAccessRoleArn": "your-service-access-ARN", "CsvRowDelimiter":
 "\n",
"CsvDelimiter": ",", "BucketFolder": "your-bucket-folder",
"BucketName": "your-bucket-name",
"EncryptionMode": "SSE_KMS",
"ServerSideEncryptionKmsKeyId": "arn:aws:kms:us-
east-1:111122223333:key/72abb6fb-1e49-4ac1-9aed-c803dfcc0480"}'
```

Ici, l'objet JSON spécifié par l'option `--s3-settings` définit deux paramètres. Le premier est le paramètre `EncryptionMode` avec la valeur `SSE_KMS`. L'autre est un paramètre `ServerSideEncryptionKmsKeyId` avec la valeur `arn:aws:kms:us-east-1:111122223333:key/72abb6fb-1e49-4ac1-9aed-c803dfcc0480`. Cette valeur est un Amazon Resource Name (ARN) pour votre clé KMS personnalisée. Pour une cible S3, vous spécifiez également des paramètres supplémentaires. Ces paramètres identifient le rôle d'accès au serveur, fournissent des délimiteurs pour le format de stockage d'objet CSV par défaut et attribuent l'emplacement et le nom du compartiment pour stocker les objets cibles S3.

Par défaut, le chiffrement de données S3 a lieu à l'aide du chiffrement côté serveur S3. Pour la cible S3 de l'exemple précédent, cela équivaut à spécifier ses paramètres de point de terminaison, comme dans l'exemple suivant.

```
aws dms create-endpoint --endpoint-identifier s3-target-endpoint --engine-name s3 --
endpoint-type target
--s3-settings '{"ServiceAccessRoleArn": "your-service-access-ARN", "CsvRowDelimiter":
 "\n",
"CsvDelimiter": ",", "BucketFolder": "your-bucket-folder",
"BucketName": "your-bucket-name",
"EncryptionMode": "SSE_S3"}'
```

Pour de plus amples informations sur l'utilisation du chiffrement côté serveur S3, veuillez consulter Protection des données à l'aide du chiffrement côté serveur.

> **Note**
>
> Vous pouvez également utiliser la commande CLI `modify-endpoint` pour modifier la valeur du paramètre `EncryptionMode` d'un point de terminaison existant de `SSE_KMS` à `SSE_S3`. Vous ne pouvez pas modifier la valeur `EncryptionMode` de `SSE_S3` à `SSE_KMS`.

Paramètres pour l'utilisation de fichiers .parquet en vue du stockage d'objets cibles S3

Le format par défaut pour la création d'objets cibles S3 est le fichier .csv. Les exemples suivants illustrent certains paramètres de point de terminaison pour spécifier des fichiers .parquet comme format de création d'objets cibles S3. Vous pouvez spécifier le format des fichiers .parquet avec toutes les valeurs par défaut, comme dans l'exemple suivant.

```
aws dms create-endpoint --endpoint-identifier s3-target-endpoint --engine-name s3 --
endpoint-type target
--s3-settings '{"ServiceAccessRoleArn": "your-service-access-ARN", "DataFormat":
 "parquet"}'
```

Ici, le paramètre `DataFormat` est défini sur `parquet` pour activer le format avec toutes les valeurs par défaut de S3. Ces valeurs par défaut comprennent un encodage par dictionnaire (`"EncodingType: "rle-dictionary"`) qui utilise une combinaison d'encodage par compression de bits et longueur d'exécution pour stocker de manière plus efficace les valeurs répétitives.

Vous pouvez ajouter des paramètres supplémentaires pour les options autres que les valeurs par défaut, comme dans l'exemple suivant.

```
aws dms create-endpoint --endpoint-identifier s3-target-endpoint --engine-name s3 --
endpoint-type target
--s3-settings '{"ServiceAccessRoleArn": "your-service-access-ARN", "BucketFolder":
 "your-bucket-folder",
"BucketName": "your-bucket-name", "CompressionType": "GZIP", "DataFormat": "parquet",
 "EncodingType: "plain-dictionary", "DictPageSizeLimit": 3,072,000,
"EnableStatistics": false }'
```

Ici, en plus des paramètres requis pour plusieurs options de compartiment S3 standard et le paramètre `DataFormat`, les paramètres de fichier .parquet supplémentaires suivants sont définis :

- `EncodingType` : permet de définir un encodage par dictionnaire (`plain-dictionary`) qui stocke les valeurs rencontrées dans chaque colonne dans un bloc par colonne de la page de dictionnaire.
- `DictPageSizeLimit` : permet de définir une taille de page de dictionnaire de 3 Mo maximum.
- `EnableStatistics` : désactive la valeur par défaut qui permet la collecte des statistiques sur les pages et les groupes de lignes du fichier Parquet.

Capture des données de modification (CDC) avec l'ordre des transactions sur la cible S3

Par défaut, lorsque AWS DMS exécute une tâche de CDC, il stocke toutes les modifications de ligne journalisées dans la base de données (ou les bases de données) source dans un ou plusieurs fichiers pour chaque table. Chaque ensemble de fichiers contenant des modifications pour la même table réside dans un répertoire cible unique associé à cette table. AWS DMS crée autant de répertoires cibles que de tables de base de données migrées vers le point de terminaison cible Amazon S3. Les fichiers sont stockés sur la cible S3 dans ces répertoires sans tenir compte de l'ordre des transactions. Pour plus d'informations sur les conventions de dénomination des fichiers, le contenu des données et le format, consultez Utilisation d'Amazon S3 en tant que cible pour AWS Database Migration Service.

Pour capturer les modifications de la base de données source de sorte à capturer également l'ordre des transactions, vous pouvez spécifier les paramètres de point de terminaison S3 qui indiquent à AWS DMS de stocker les modifications de ligne pour toutes les tables de base de données dans un ou plusieurs fichiers .csv créés en fonction de la taille de transaction. Ces fichiers de transaction .csv contiennent toutes les modifications de ligne répertoriées séquentiellement dans l'ordre des transactions pour toutes les tables impliquées dans chaque transaction. Ces fichiers de transaction résident ensemble dans un répertoire de transactions unique que vous spécifiez également sur la cible S3. Dans chaque fichier de transaction, l'opération de transaction et l'identité de la base de données et de la table source pour chaque modification de ligne sont stockées dans le cadre des données de ligne, comme suit.

```
operation,table_name,database_schema_name,field_value,...
```

Ici, *`operation`* est l'opération de transaction sur la ligne modifiée, *`table_name`* est le nom de la table de base de données où la ligne est modifiée, *`database_schema_name`* est le nom du schéma de base de données dans lequel se trouve la table et *`field_value`* est la première d'une ou de plusieurs valeurs de champ qui spécifient les données de la ligne.

L'exemple suivant d'un fichier de transaction illustre les lignes modifiées pour une ou plusieurs transactions impliquant deux tables.

```
I,Names_03cdcad11a,rdsTempsdb,13,Daniel
U,Names_03cdcad11a,rdsTempsdb,23,Kathy
D,Names_03cdcad11a,rdsTempsdb,13,Cathy
I,Names_6d152ce62d,rdsTempsdb,15,Jane
I,Names_6d152ce62d,rdsTempsdb,24,Chris
I,Names_03cdcad11a,rdsTempsdb,16,Mike
```

Ici, l'opération de transaction sur chaque ligne est indiquée par `I` (insérer), `U` (mettre à jour) ou `D` (supprimer) dans la première colonne. Le nom de la table est la valeur de la deuxième colonne (par exemple, `Names_03cdcad11a`). Le nom du schéma de base de données est la valeur de la troisième colonne (par exemple, `rdsTempsdb`). Et les colonnes restantes sont remplies avec vos propres données de ligne (par exemple, `13,Daniel`).

En outre, AWS DMS nomme les fichiers de transaction qu'il crée sur la cible Amazon S3 à l'aide d'un horodatage conformément à la convention de dénomination suivante.

```
CDC_TXN-timestamp.csv
```

Ici, *`timestamp`* est l'heure à laquelle le fichier de transaction a été créé, comme dans l'exemple suivant.

```
CDC_TXN-20201117153046033.csv
```

Cet horodatage dans le nom du fichier garantit que les fichiers de transaction sont créés et répertoriés dans l'ordre des transactions lorsque vous les répertoriez dans leur répertoire de transactions.

> **Note**
>
> Lors de la capture des modifications de données dans l'ordre des transactions, AWS DMS stocke toujours les modifications de ligne dans des fichiers .csv, quelle que soit la valeur du paramètre S3 `DataFormat` sur la cible. AWS DMS n'enregistre pas les modifications de données dans l'ordre des transactions à l'aide de fichiers .parquet.

Pour contrôler la fréquence des écritures sur une cible Amazon S3 lors d'une tâche de réplication de données, vous pouvez configurer les paramètres `CdcMaxBatchInterval` et `CdcMinFileSize`. Cela peut améliorer les performances lors de l'analyse des données sans aucune surcharge supplémentaire. Pour de plus amples informations, veuillez consulter Paramètres de point de terminaison lors de l'utilisation d'Amazon S3 en tant que cible pour AWS DMS.

Pour indiquer à AWS DMS de stocker toutes les modifications de ligne dans l'ordre des transactions

1. Définissez le paramètre S3 `PreserveTransactions` sur la cible sur `true`.
2. Définissez le paramètre S3 `CdcPath` sur la cible sur un chemin de dossier relatif dans lequel vous souhaitez que AWS DMS stocke les fichiers de transaction .csv.

   AWS DMS crée ce chemin soit dans le compartiment cible et le répertoire de travail S3 par défaut, soit dans le compartiment et le dossier de compartiment que vous spécifiez à l'aide des paramètres S3 `BucketName` et `BucketFolder` sur la cible.

Indication des opérations de base de données source dans des données S3 migrées

Lorsque AWS DMS migre les enregistrements vers une cible S3, un champ supplémentaire peut être créé dans chaque enregistrement migré. Ce champ supplémentaire indique l'opération appliquée à l'enregistrement au niveau de la base de données source. La façon dont AWS DMS crée et définit ce premier champ dépend du type de tâche de migration et des valeurs de `includeOpForFullLoad`, `cdcInsertsOnly` et `cdcInsertsAndUpdates`.

Pour un chargement complet, lorsque la valeur de `includeOpForFullLoad` est `true`, AWS DMS crée toujours un premier champ supplémentaire dans chaque enregistrement .csv. Ce champ contient la lettre I (INSERT) pour indiquer que la ligne a été insérée au niveau de la base de données source. Pour un chargement CDC, lorsque la valeur de `cdcInsertsOnly` est `false` (valeur par défaut), AWS DMS crée également toujours un premier champ supplémentaire dans chaque enregistrement .csv ou .parquet. Ce champ contient la lettre I (INSERT), U (UPDATE) ou D (DELETE) pour indiquer si la ligne a été insérée, mise à jour ou supprimée au niveau de la base de données source.

Dans le tableau suivant, vous pouvez voir comment les paramètres des attributs `includeOpForFullLoad` et `cdcInsertsOnly` fonctionnent ensemble pour modifier la configuration des enregistrements migrés.

Avec ces valeurs de paramètres		DMS définit les enregistrements cibles comme suit pour la sortie .csv et .parquet	
includeOpForFullLoad	cdcInsertsOnly	Pour un chargement complet	Pour un chargement CDC
`true`	`true`	Ajout de la valeur du premier champ définie sur `I`	Ajout de la valeur du premier champ définie sur `I`
`false`	`false`	Aucun champ ajouté	Ajout de la valeur du premier champ définie sur `I`, `U` ou `D`
`false`	`true`	Aucun champ ajouté	Aucun champ ajouté
`true`	`false`	Ajout de la valeur du premier champ définie sur `I`	Ajout de la valeur du premier champ définie sur `I`, `U` ou `D`

Lorsque `includeOpForFullLoad` et `cdcInsertsOnly` sont définis sur la même valeur, les enregistrements cibles sont définis en fonction de l'attribut qui contrôle les paramètres d'enregistrement pour le type de migration actuel. Cet attribut est `includeOpForFullLoad` pour les chargements complets et `cdcInsertsOnly` pour les chargements CDC.

Lorsque `includeOpForFullLoad` et `cdcInsertsOnly` sont définis sur des valeurs différentes, AWS DMS assure la cohérence des paramètres de l'enregistrement pour les chargements CDC et les chargements complets. Pour ce faire, il met en conformité les paramètres d'enregistrement pour un chargement CDC avec ceux du chargement complet précédent spécifié par `includeOpForFullLoad`.

En d'autres termes, supposons qu'un chargement complet soit défini de façon à ajouter un premier champ pour indiquer un enregistrement inséré. Dans ce cas, un chargement CDC suivant est défini de façon à ajouter un premier champ qui indique un enregistrement inséré, mis à jour ou supprimé (selon l'enregistrement au niveau de la source). En revanche, supposons qu'un chargement complet soit défini de façon à ne pas ajouter un premier champ pour indiquer un enregistrement inséré. Dans ce cas, un chargement CDC est également défini de façon à ne pas ajouter un premier champ pour

chaque enregistrement, quelle que soit l'opération d'enregistrement correspondante au niveau de la source.

De même, la façon dont DMS crée et définit un premier champ supplémentaire dépend des paramètres de `includeOpForFullLoad` et `cdcInsertsAndUpdates`. Dans le tableau suivant, vous pouvez voir comment les paramètres des attributs `includeOpForFullLoad` et `cdcInsertsAndUpdates` fonctionnent ensemble pour modifier la configuration des enregistrements migrés sous ce format.

Avec ces valeurs de paramètres		DMS définit les enregistrements cibles comme suit pour la sortie .csv	
includeOpForFullLoad	cdcInsertsAndUpdates	Pour un chargement complet	Pour un chargement CDC
`true`	`true`	Ajout de la valeur du premier champ définie sur `I`	Ajout de la valeur du premier champ définie sur `I` ou `U`
`false`	`false`	Aucun champ ajouté	Ajout de la valeur du premier champ définie sur `I`, `U` ou `D`
`false`	`true`	Aucun champ ajouté	Ajout de la valeur du premier champ définie sur `I` ou `U`
`true`	`false`	Ajout de la valeur du premier champ définie sur `I`	Ajout de la valeur du premier champ définie sur `I`, `U` ou `D`

Types de données cibles pour S3 Parquet

Le tableau suivant présente les types de données cibles Parquet qui sont pris en charge lorsque vous utilisez AWS DMS et le mappage par défaut à partir des types de données AWS DMS.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Type de données AWS DMS	fType de données S3 Parquet
BYTES	BINARY
DATE	DATE32
TIME	TIME32
DATETIME	TIMESTAMP
INT1	INT8
INT2	INT16
INT4	INT32
INT8	INT64
NUMERIC	DECIMAL
REAL4	FLOAT
REAL8	DOUBLE
STRING	STRING
UINT1	UINT8
UINT2	UINT16
UINT4	UINT32
UINT8	UINT64
WSTRING	STRING
BLOB	BINARY
NCLOB	STRING
CLOB	STRING

Type de données AWS DMS	fType de données S3 Parquet
BOOLEAN	BOOL

Utilisation d'une base de données Amazon DynamoDB en tant que cible pour AWS Database Migration Service

Vous pouvez utiliser AWS DMS pour migrer les données vers une table Amazon DynamoDB. Amazon DynamoDB est un service de base de données NoSQL entièrement géré qui offre des performances exceptionnelles et prévisibles en termes de rapidité et de capacité de mise à l'échelle. AWS DMS prend en charge l'utilisation d'une base de données relationnelle ou de MongoDB en tant que source.

Dans DynamoDB, les tables, les éléments et les attributs sont les principaux composants que vous utilisez. Une table est une collection d'éléments et chaque élément est une collection d'attributs. DynamoDB utilise les clés primaires, appelées clés de partition, afin d'identifier de façon unique chaque élément d'une table. Vous pouvez également utiliser des clés et des index secondaires pour fournir une plus grande flexibilité d'interrogation.

Vous utilisez le mappage d'objet afin de migrer vos données d'une base de données source vers une table DynamoDB cible. Le mappage d'objet vous permet de déterminer l'emplacement des données sources dans la cible.

Quand AWS DMS crée les tables sur un point de terminaison cible DynamoDB, il crée autant de tables que dans le point de terminaison de la base de données source. AWS DMS définit également plusieurs valeurs de paramètre DynamoDB. Le coût de création de la table dépend de la quantité de données et du nombre de tables à migrer.

> **Note**
>
> L'option Mode SSL de la console AWS DMS ou de l'API ne s'applique pas à certains flux de données et services NoSQL tels que Kinesis et DynamoDB. Étant donné qu'ils sont sécurisés par défaut, AWS DMS indique que le paramètre du mode SSL est None (SSL Mode=None). Vous n'avez pas besoin de fournir de configuration supplémentaire pour que votre point de terminaison utilise le protocole SSL. Par exemple, lorsque DynamoDB est utilisé comme point de terminaison cible, il est sécurisé par défaut. Tous les appels d'API à DynamoDB utilisent le protocole SSL. Il n'est donc pas nécessaire d'ajouter une option SSL supplémentaire dans

le point de terminaison AWS DMS. Vous pouvez placer et récupérer des données en toute sécurité via des points de terminaison SSL à l'aide du protocole HTTPS, utilisé par AWS DMS par défaut lors de la connexion à une base de données DynamoDB.

Pour augmenter la vitesse du transfert, AWS DMS prend en charge un chargement complet multithread sur une instance cible DynamoDB. DMS prend en charge ce traitement multithread avec des paramètres de tâche, notamment les suivants :

- `MaxFullLoadSubTasks` : utilisez cette option pour indiquer le nombre maximal de tables sources à charger en parallèle. DMS charge chaque table dans sa table cible DynamoDB correspondante à l'aide d'une sous-tâche dédiée. La valeur par défaut est 8. La valeur maximale est 49.
- `ParallelLoadThreads` : utilisez cette option pour spécifier le nombre de threads utilisés par AWS DMS pour charger chaque table dans sa table cible DynamoDB. La valeur par défaut est 0 (à thread unique). La valeur maximale est 200. Vous pouvez demander une augmentation de cette limite maximale.

  > **Note**
  >
  > DMS attribue chaque segment d'une table à son propre thread pour le chargement. Par conséquent, définissez `ParallelLoadThreads` sur le nombre maximal de segments que vous spécifiez pour une table de la source.

- `ParallelLoadBufferSize` : utilisez cette option pour spécifier le nombre maximal d'enregistrements à stocker dans la mémoire tampon utilisée par les threads de chargement parallèles pour charger les données dans la cible DynamoDB. La valeur par défaut est 50. La valeur maximale est 1 000. Utilisez ce paramètre avec `ParallelLoadThreads`. `ParallelLoadBufferSize` est valide uniquement dans le cas de plusieurs threads.
- Paramètres de mappage de table pour des tables individuelles : utilisez des règles `table-settings` pour identifier les tables individuelles de la source que vous souhaitez charger en parallèle. Utilisez également ces règles pour spécifier comment segmenter les lignes de chaque table pour le chargement multithread. Pour de plus amples informations, veuillez consulter Règles des paramètres de table et de collection et opérations.

> **Note**
>
> Lorsque AWS DMS définit les valeurs des paramètres DynamoDB pour une tâche de migration, la valeur par défaut du paramètre RCU (Unités de capacité de lecture) est définie sur 200.
> La valeur du paramètre WCU (Write Capacity Units) est également définie, mais sa valeur dépend de plusieurs autres paramètres :
>
> - La valeur par défaut du paramètre WCU est 200.
> - Si le paramètre de tâche `ParallelLoadThreads` est défini sur une valeur supérieure à 1 (0 par défaut), le paramètre WCU est défini à 200 fois la valeur `ParallelLoadThreads`.
> - Les frais d'utilisation standard d'AWS DMS s'appliquent aux ressources que vous utilisez.

Migration à partir d'une base de données relationnelle vers une table DynamoDB

AWS DMS prend en charge la migration de données vers les types de données scalaires DynamoDB. Lors d'une migration depuis une base de données relationnelle telle qu'Oracle ou MySQL vers DynamoDB, vous pouvez restructurer la manière dont vous stockez les données.

AWS DMS prend actuellement en charge la restructuration d'une table unique vers une table unique d'attributs de types scalaires DynamoDB. Si vous migrez des données vers DynamoDB depuis une table de base de données relationnelle, vous prenez les données d'une table et les reformatez en attributs de types de données scalaires DynamoDB. Ces attributs peuvent accepter des données de plusieurs colonnes et vous pouvez mapper directement une colonne à un attribut.

AWS DMS prend en charge les types de données scalaires DynamoDB suivants :

- Chaîne
- Nombre
- Booléen

> **Note**
>
> Les données NULL provenant de la source sont ignorées dans la cible.

Prérequis pour l'utilisation de DynamoDB en tant que cible pour AWS Database Migration Service

Avant de commencer à utiliser une base de données DynamoDB en tant que cible pour AWS DMS, veillez à créer un rôle IAM. Ce rôle IAM doit permettre à AWS DMS d'assumer et d'accorder l'accès aux tables DynamoDB qui constituent la cible de la migration. L'ensemble d'autorisations d'accès minimum est indiqué dans la stratégie IAM suivante.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "",
         "Effect": "Allow",
         "Principal": {
            "Service": "dms.amazonaws.com"
         },
         "Action": "sts:AssumeRole"
      }
   ]
}
```

Le rôle que vous utilisez pour la migration vers DynamoDB doit bénéficier des autorisations suivantes.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Action": [
        "dynamodb:PutItem",
        "dynamodb:CreateTable",
        "dynamodb:DescribeTable",
        "dynamodb:DeleteTable",
        "dynamodb:DeleteItem",
        "dynamodb:UpdateItem"
    ],
    "Resource": [
    "arn:aws:dynamodb:us-west-2:account-id:table/name1",
    "arn:aws:dynamodb:us-west-2:account-id:table/OtherName*",
```

```
    "arn:aws:dynamodb:us-west-2:account-id:table/awsdms_apply_exceptions",
    "arn:aws:dynamodb:us-west-2:account-id:table/awsdms_full_load_exceptions"
    ]
    },
    {
        "Effect": "Allow",
        "Action": [
        "dynamodb:ListTables"
        ],
        "Resource": "*"
    }
    ]
}
```

Limitations de l'utilisation de DynamoDB en tant que cible pour AWS Database Migration Service

Les limitations suivantes s'appliquent à l'utilisation de DynamoDB en tant que cible :

- DynamoDB limite la précision du type de données Number à 38 places. Stockez tous les types de données en tant que chaîne avec une meilleure précision. Vous devez le spécifier explicitement à l'aide de la fonction de mappage d'objet.
- Étant donné que DynamoDB ne possède pas de type de données Date, les données utilisant le type de données Date sont converties en chaînes.
- DynamoDB n'autorise pas les mises à jour des attributs de la clé primaire. Cette restriction est importante lors de l'utilisation de la réplication continue avec la capture des données modifiées (CDC), car elle peut entraîner des données non souhaitées dans la cible. En fonction de la façon dont l'objet est mappé, une opération CDC qui met à jour la clé primaire peut réaliser l'une des deux actions suivantes. Elle peut échouer ou insérer un nouvel élément avec la clé primaire mise à jour et des données incomplètes.
- AWS DMS prend uniquement en charge la réplication de tables avec des clés primaires non composites. Une exception s'applique si vous spécifiez un mappage d'objet pour la table cible avec une clé de partition ou une clé de tri personnalisée, ou les deux.
- AWS DMS ne prend pas en charge les données LOB à moins qu'il s'agisse de données CLOB. AWS DMS convertit les données CLOB en une chaîne DynamoDB lors de la migration des données.

- Lorsque vous utilisez DynamoDB comme cible, seule la table de contrôle Appliquer les exceptions (`dmslogs.awsdms_apply_exceptions`) est prise en charge. Pour plus d'informations sur les tables de contrôle, consultez Paramètres de tâche de la table de contrôle.
- AWS DMS ne prend pas en charge le paramètre de tâche `TargetTablePrepMode=TRUNCATE_BEFORE_LOAD` pour DynamoDB en tant que cible.
- AWS DMS ne prend pas en charge le paramètre de tâche `TaskRecoveryTableEnabled` pour DynamoDB en tant que cible.

Utilisation du mappage d'objet pour migrer les données vers DynamoDB

AWS DMS utilise les règles de mappage de tables pour mapper les données depuis la source vers la table DynamoDB cible. Pour mapper des données vers une cible DynamoDB, utilisez un type de règle de mappage de tables dénommé object-mapping. Le mappage d'objet vous permet de définir le nom des attributs et les données qui doivent être migrées vers ceux-ci. Des règles de sélection doivent être définies lorsque vous utilisez le mappage d'objet.

DynamoDB ne dispose pas d'une structure prédéfinie autre qu'une clé de partition et une clé de tri facultative. Si vous disposez d'une clé primaire non composite, AWS DMS l'utilise. Si vous avez une clé primaire composite ou si vous souhaitez utiliser une clé de tri, définissez ces clés et les autres attributs dans votre table DynamoDB cible.

Pour créer une règle de mappage d'objet, vous spécifiez le `rule-type` comme mappage d'objet. Cette règle spécifie le type de mappage d'objet que vous souhaitez utiliser.

La structure de la règle est la suivante :

```
{ "rules": [
    {
      "rule-type": "object-mapping",
      "rule-id": "<id>",
      "rule-name": "<name>",
      "rule-action": "<valid object-mapping rule action>",
      "object-locator": {
      "schema-name": "<case-sensitive schema name>",
      "table-name": ""
      },
      "target-table-name": "<table_name>"
    }
  ]
```

```
}
```

AWS DMS prend actuellement en charge `map-record-to-record` et `map-record-to-document` comme les seules valeurs valides pour le paramètre `rule-action`. Ces valeurs spécifient l'action par défaut d'AWS DMS sur les enregistrements qui ne sont pas exclus de par leur appartenance à la liste des attributs `exclude-columns`. Ces valeurs n'affectent en aucune façon les mappages d'attributs.

- Vous pouvez utiliser `map-record-to-record` lors de la migration d'une base de données relationnelle vers DynamoDB. La clé primaire est utilisée depuis la base de données relationnelle comme clé de partition dans DynamoDB et crée un attribut pour chaque colonne dans la base de données source. Lorsque vous utilisez `map-record-to-record`, pour toute colonne de la table source non répertoriée dans la liste d'attributs `exclude-columns`, AWS DMS crée un attribut correspondant sur l'instance DynamoDB cible. Cet attribut est créé, que la colonne source soit ou non utilisée dans un mappage d'attribut.
- Vous utilisez `map-record-to-document` pour placer des colonnes sources dans un mappage DynamoDB unique et plat sur la cible, avec le nom d'attribut « _doc ». Lorsque vous utilisez `map-record-to-document`, AWS DMS place les données dans un attribut de mappage DynamoDB unique et plat sur la source. Cet attribut est appelé « _doc ». Ce placement s'applique à toute colonne de la table source non listée dans la liste d'attributs `exclude-columns`.

Une des manières de comprendre la différence entre les paramètres `rule-action` `map-record-to-record` et `map-record-to-document` consiste à voir les deux paramètres en action. Dans cet exemple, imaginons que vous commencez avec une ligne de table d'une base de données relationnelle, présentant la structure et les données suivantes :

FirstName	LastName	NickName	WorkAddress	WorkPhone	HomeAddress	HomePhone	income
Daniel	Sheridan	Dan	101 Main St Cambridge, MA	800-867-5309	100 Secret St, Unknownville, MA	123-456-7890	12345678

Pour migrer ces informations vers DynamoDB, vous créez des règles pour mapper les données dans un élément de la table DynamoDB. Notez les colonnes listées pour le paramètre `exclude-columns`. Ces colonnes ne sont pas directement mappées à la cible. Le mappage d'attribut est plutôt utilisé pour combiner les données en nouveaux éléments, comme par exemple lorsque FirstName et LastName sont regroupés pour former CustomerName dans la cible DynamoDB. NickName et income ne sont pas exclus.

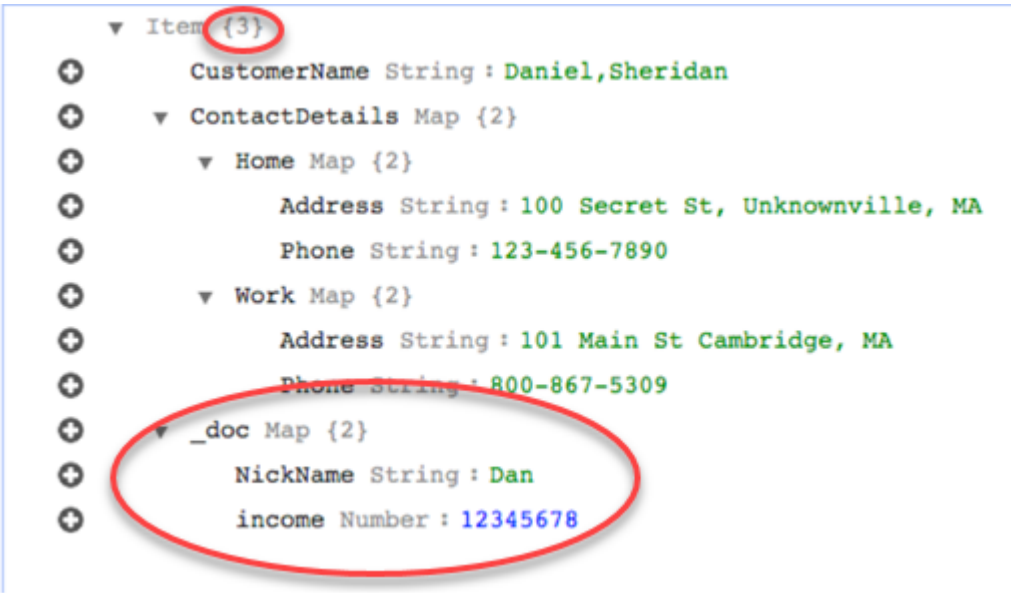
```
{
```

```
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "2",
            "rule-name": "TransformToDDB",
            "rule-action": "map-record-to-record",
            "object-locator": {
                "schema-name": "test",
                "table-name": "customer"
            },
            "target-table-name": "customer_t",
            "mapping-parameters": {
                "partition-key-name": "CustomerName",
                "exclude-columns": [
                    "FirstName",
                    "LastName",
                    "HomeAddress",
                    "HomePhone",
                    "WorkAddress",
                    "WorkPhone"
                ],
                "attribute-mappings": [
                    {
                        "target-attribute-name": "CustomerName",
                        "attribute-type": "scalar",
                        "attribute-sub-type": "string",
                        "value": "${FirstName},${LastName}"
                    },
                    {
                        "target-attribute-name": "ContactDetails",
                        "attribute-type": "document",
                        "attribute-sub-type": "dynamodb-map",
                        "value": {
                            "M": {
```

```
                                "Home": {
                                    "M": {
                                        "Address": {
                                            "S": "${HomeAddress}"
                                        },
                                        "Phone": {
                                            "S": "${HomePhone}"
                                        }
                                    }
                                },
                                "Work": {
                                    "M": {
                                        "Address": {
                                            "S": "${WorkAddress}"
                                        },
                                        "Phone": {
                                            "S": "${WorkPhone}"
                                        }
                                    }
                                }
                            }
                        }
                    ]
                }
            }
        ]
}
```

En utilisant le `rule-action` paramètre map-record-to-record, les données de NickName et income sont mappées à des éléments du même nom dans la cible DynamoDB.

Toutefois, supposons que vous utilisez les mêmes règles mais que vous remplacez le paramètre `rule-action` par map-record-to-document. Dans ce cas, les colonnes non répertoriées dans le paramètre `exclude-columns`, NickName et income, sont mappées à un élément _doc.



Utilisation d'expressions de condition personnalisées avec le mappage d'objet

Vous pouvez utiliser une fonctionnalité de DynamoDB nommée « expressions conditionnelles » pour manipuler les données qui sont écrites dans une table DynamoDB. Pour plus d'informations sur les expressions de condition dans DynamoDB, consultez Expressions de condition.

Une expression de condition est constituée des éléments suivants :

- une expression (obligatoire)

- des valeurs d'attributs d'expressions (facultatif). Spécifie une structure DynamoDB json de la valeur de l'attribut
- des noms d'attributs d'expressions (facultatif)
- options pour déterminer quand utiliser l'expression de condition (facultatif). La valeur par défaut est apply-during-cdc = false et apply-during-full-load = true

La structure de la règle est la suivante :

```
"target-table-name": "customer_t",
      "mapping-parameters": {
        "partition-key-name": "CustomerName",
        "condition-expression": {
          "expression":"<conditional expression>",
          "expression-attribute-values": [
              {
                "name":"<attribute name>",
                "value":<attribute value>
              }
          ],
          "apply-during-cdc":<optional Boolean value>,
          "apply-during-full-load": <optional Boolean value>
        }
```

L'exemple suivant illustre les sections utilisées pour l'expression de condition.

```
{
  "rules": [
    {
      "rule-type": "object-mapping",
      "rule-id": "1",
      "rule-name": "TransformToDDB",
      "rule-action": "map-record-to-record",
      "object-locator": {
        "schema-name": "test",
        "table-name": "customer",

      },
      "target-table-name": "customer_t",
      "mapping-parameters": {
        "partition-key-name": "CustomerName",
        "condition-expression": {
          "expression":"attribute_not_exists(version) or version <= :record_version",
          "expression-attribute-values": [
              {
                "name":":record_version",
                "value":{"N":"${version}"}
              }
          ],
          "apply-during-cdc":true,
          "apply-during-full-load": true
        }
        "attribute-mappings": [
          {
            "target-attribute-name": "CustomerName",
            "attribute-type": "scalar",
            "attribute-sub-type": "string",
            "value": "${FirstName},${LastName}"
          }
        ]
      }
    }
  ]
}
```

Utilisation du mappage d'attribut avec le mappage d'objet

La mappage d'attribut vous permet de spécifier une chaîne modèle, en utilisant les noms de la colonne source pour restructurer les données sur la cible. Aucun formatage n'est effectué, autre que ce qui est spécifié par l'utilisateur dans le modèle.

L'exemple suivant illustre la structure de la base de données source et la structure de cible DynamoDB souhaitée. La structure de la source est illustrée en premier, dans ce cas, une base de données Oracle, puis la structure souhaitée des données dans DynamoDB. L'exemple se termine avec le JSON utilisé pour créer la structure cible souhaitée.

La structure des données Oracle est la suivante :

First	Las	St	Home/ ss	HomeF	WorkAddre ss	Work	DateOfBirth
Clé primaire				N/A			
Ran	Mar	5	221B Baker Street	123456 0	31 Spooner Street, Quahog	9876 0	02/29/1988

La structure des données DynamoDB est la suivante :

Custor ame	StoreId	ContactDetails	DateOfBirth
Clé de partition	Clé de tri	N/A	
Randy sh	5	{ "Name": "Randy", "Home": { "Address": "221B Baker Street", "Phone": 1234567890 }, "Work": { "Address": "31 Spooner Street, Quahog", "Phone": 9876541230 } }	02/29/1988

Le JSON suivant illustre le mappage d'objet et le mappage de colonne utilisés pour parvenir à la structure DynamoDB :

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "2",
            "rule-name": "TransformToDDB",
            "rule-action": "map-record-to-record",
            "object-locator": {
                "schema-name": "test",
                "table-name": "customer"
            },
            "target-table-name": "customer_t",
            "mapping-parameters": {
                "partition-key-name": "CustomerName",
                "sort-key-name": "StoreId",
                "exclude-columns": [
                    "FirstName",
                    "LastName",
                    "HomeAddress",
                    "HomePhone",
                    "WorkAddress",
                    "WorkPhone"
                ],
                "attribute-mappings": [
                    {
                        "target-attribute-name": "CustomerName",
                        "attribute-type": "scalar",
                        "attribute-sub-type": "string",
                        "value": "${FirstName},${LastName}"
                    },
                    {
                        "target-attribute-name": "StoreId",
```

```
                        "attribute-type": "scalar",
                        "attribute-sub-type": "string",
                        "value": "${StoreId}"
                    },
                    {
                        "target-attribute-name": "ContactDetails",
                        "attribute-type": "scalar",
                        "attribute-sub-type": "string",
                        "value": "{\"Name\":\"${FirstName}\",\"Home\":{\"Address
\":\"${HomeAddress}\",\"Phone\":\"${HomePhone}\"}, \"Work\":{\"Address\":
\"${WorkAddress}\",\"Phone\":\"${WorkPhone}\"}}"
                    }
                ]
            }
        }
    ]
}
```

Une autre façon d'utiliser le mappage de colonne consiste à utiliser le format DynamoDB comme type de document. L'exemple de code suivant utilise dynamodb-map comme `attribute-sub-type` pour la mappage d'attribut.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "2",
            "rule-name": "TransformToDDB",
            "rule-action": "map-record-to-record",
            "object-locator": {
                "schema-name": "test",
```

```
            "table-name": "customer"
        },
        "target-table-name": "customer_t",
        "mapping-parameters": {
            "partition-key-name": "CustomerName",
            "sort-key-name": "StoreId",
            "exclude-columns": [
                "FirstName",
                "LastName",
                "HomeAddress",
                "HomePhone",
                "WorkAddress",
                "WorkPhone"
            ],
            "attribute-mappings": [
                {
                    "target-attribute-name": "CustomerName",
                    "attribute-type": "scalar",
                    "attribute-sub-type": "string",
                    "value": "${FirstName},${LastName}"
                },
                {
                    "target-attribute-name": "StoreId",
                    "attribute-type": "scalar",
                    "attribute-sub-type": "string",
                    "value": "${StoreId}"
                },
                {
                    "target-attribute-name": "ContactDetails",
                    "attribute-type": "document",
                    "attribute-sub-type": "dynamodb-map",
                    "value": {
                      "M": {
                        "Name": {
                          "S": "${FirstName}"
                        },
                        "Home": {
                                "M": {
                                    "Address": {
                                        "S": "${HomeAddress}"
                                    },
                                    "Phone": {
                                        "S": "${HomePhone}"
                                    }
```

```
                                    }
                                },
                                "Work": {
                                    "M": {
                                        "Address": {
                                            "S": "${WorkAddress}"
                                        },
                                        "Phone": {
                                            "S": "${WorkPhone}"
                                        }
                                    }
                                }
                            }
                        }
                    }
                ]
            }
        }
    ]
}
```

Au lieu d'utiliser `dynamodb-map`, vous pouvez utiliser `dynamodb-list` comme attribute-sub-type pour le mappage d'attribut, comme indiqué dans l'exemple suivant.

```
{
"target-attribute-name": "ContactDetailsList",
"attribute-type": "document",
"attribute-sub-type": "dynamodb-list",
"value": {
    "L": [
            {
                "N": "${FirstName}"
            },
            {
                "N": "${HomeAddress}"
            },
            {
                "N": "${HomePhone}"
            },
            {
                "N": "${WorkAddress}"
```

```
            },
            {
                "N": "${WorkPhone}"
            }
        ]
    }
}
```

Exemple 1 : utilisation du mappage d'attribut avec le mappage d'objet

L'exemple suivant migre les données depuis deux tables de base de données MySQL, nfl_data et sport_team, vers deux tables DynamoDB dénommées NFLTeams et SportTeams. La structure des tables et le JSON utilisés pour mapper les données depuis les tables de base de données MySQL vers les tables DynamoDB sont les suivants :

La structure de la table de base de données MySQL nfl_data est indiquée ci-dessous :

```
mysql> desc nfl_data;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Position      | varchar(5)  | YES  |     | NULL    |       |
| player_number | smallint(6) | YES  |     | NULL    |       |
| Name          | varchar(40) | YES  |     | NULL    |       |
| status        | varchar(10) | YES  |     | NULL    |       |
| stat1         | varchar(10) | YES  |     | NULL    |       |
| stat1_val     | varchar(10) | YES  |     | NULL    |       |
| stat2         | varchar(10) | YES  |     | NULL    |       |
| stat2_val     | varchar(10) | YES  |     | NULL    |       |
| stat3         | varchar(10) | YES  |     | NULL    |       |
| stat3_val     | varchar(10) | YES  |     | NULL    |       |
| stat4         | varchar(10) | YES  |     | NULL    |       |
| stat4_val     | varchar(10) | YES  |     | NULL    |       |
| team          | varchar(10) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```

La structure de la table de base de données MySQL sport_team est indiquée ci-dessous :

```
mysql> desc sport_team;
+--------------------------+-------------+------+-----+---------+----------------+
| Field                    | Type        | Null | Key | Default | Extra          |
+--------------------------+-------------+------+-----+---------+----------------+
| id                       | mediumint(9) | NO   | PRI | NULL    | auto_increment |
| name                     | varchar(30)  | NO   |     | NULL    |                |
| abbreviated_name         | varchar(10)  | YES  |     | NULL    |                |
| home_field_id            | smallint(6)  | YES  | MUL | NULL    |                |
| sport_type_name          | varchar(15)  | NO   | MUL | NULL    |                |
| sport_league_short_name  | varchar(10)  | NO   |     | NULL    |                |
| sport_division_short_name | varchar(10)  | YES  |     | NULL    |                |
```

Les règles de mappage de table utilisées pour mapper les deux tables vers les deux tables DynamoDB sont indiquées ci-dessous :

```
{
  "rules":[
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "dms_sample",
        "table-name": "nfl_data"
      },
      "rule-action": "include"
    },
    {
      "rule-type": "selection",
      "rule-id": "2",
      "rule-name": "2",
      "object-locator": {
        "schema-name": "dms_sample",
        "table-name": "sport_team"
      },
      "rule-action": "include"
    },
    {
      "rule-type":"object-mapping",
      "rule-id":"3",
      "rule-name":"MapNFLData",
```

```
      "rule-action":"map-record-to-record",
      "object-locator":{
        "schema-name":"dms_sample",
        "table-name":"nfl_data"
      },
      "target-table-name":"NFLTeams",
      "mapping-parameters":{
        "partition-key-name":"Team",
        "sort-key-name":"PlayerName",
        "exclude-columns": [
          "player_number", "team", "name"
        ],
        "attribute-mappings":[
          {
            "target-attribute-name":"Team",
            "attribute-type":"scalar",
            "attribute-sub-type":"string",
            "value":"${team}"
          },
          {
            "target-attribute-name":"PlayerName",
            "attribute-type":"scalar",
            "attribute-sub-type":"string",
            "value":"${name}"
          },
          {
            "target-attribute-name":"PlayerInfo",
            "attribute-type":"scalar",
            "attribute-sub-type":"string",
            "value":"{\"Number\": \"${player_number}\",\"Position\": \"${Position}\",
\"Status\": \"${status}\",\"Stats\": {\"Stat1\": \"${stat1}:${stat1_val}\",\"Stat2\":
 \"${stat2}:${stat2_val}\",\"Stat3\": \"${stat3}:${
stat3_val}\",\"Stat4\": \"${stat4}:${stat4_val}\"}"
          }
        ]
      }
    },
    {
      "rule-type":"object-mapping",
      "rule-id":"4",
      "rule-name":"MapSportTeam",
      "rule-action":"map-record-to-record",
      "object-locator":{
        "schema-name":"dms_sample",
```

```
        "table-name":"sport_team"
      },
      "target-table-name":"SportTeams",
      "mapping-parameters":{
        "partition-key-name":"TeamName",
        "exclude-columns": [
          "name", "id"
        ],
        "attribute-mappings":[
          {
            "target-attribute-name":"TeamName",
            "attribute-type":"scalar",
            "attribute-sub-type":"string",
            "value":"${name}"
          },
          {
            "target-attribute-name":"TeamInfo",
            "attribute-type":"scalar",
            "attribute-sub-type":"string",
            "value":"{\"League\": \"${sport_league_short_name}\",\"Division\":
 \"${sport_division_short_name}\"}"
          }
        ]
      }
    }
  ]
}
```

L'exemple de sortie pour la table DynamoDB NFLTeams est indiqué ci-dessous :

```
  "PlayerInfo": "{\"Number\": \"6\",\"Position\": \"P\",\"Status\": \"ACT\",\"Stats\":
 {\"Stat1\": \"PUNTS:73\",\"Stat2\": \"AVG:46\",\"Stat3\": \"LNG:67\",\"Stat4\": \"IN
 20:31\"}",
  "PlayerName": "Allen, Ryan",
  "Position": "P",
  "stat1": "PUNTS",
  "stat1_val": "73",
  "stat2": "AVG",
  "stat2_val": "46",
  "stat3": "LNG",
  "stat3_val": "67",
```

```
  "stat4": "IN 20",
  "stat4_val": "31",
  "status": "ACT",
  "Team": "NE"
}
```

L'exemple de sortie de la table SportsTeams DynamoDB est indiqué ci-dessous :

```
{
  "abbreviated_name": "IND",
  "home_field_id": 53,
  "sport_division_short_name": "AFC South",
  "sport_league_short_name": "NFL",
  "sport_type_name": "football",
  "TeamInfo": "{\"League\": \"NFL\",\"Division\": \"AFC South\"}",
  "TeamName": "Indianapolis Colts"
}
```

Types de données cibles pour DynamoDB

Le point de terminaison DynamoDB pour AWS DMS prend en charge la plupart des types de données DynamoDB. Le tableau suivant présente les types de données cibles Amazon AWS DMS qui sont pris en charge lorsque vous utilisez AWS DMS et le mappage par défaut à partir des types de données AWS DMS.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Lorsqu'AWS DMS migre des données depuis des bases de données hétérogènes, nous mappons les types de données depuis la base de données source vers les types de données intermédiaires, dénommés types de données AWS DMS. Nous mappons ensuite les types de données intermédiaires vers les types de données cibles. Le tableau suivant illustre chaque type de données AWS DMS et le type de données qu'il mappe dans DynamoDB :

Type de données AWS DMS	Type de données DynamoDB
Chaîne	Chaîne

Type de données AWS DMS	Type de données DynamoDB
WString	Chaîne
Booléen	Booléen
Date	Chaîne
DateTime	Chaîne
INT1	Nombre
INT2	Nombre
INT4	Nombre
INT8	Nombre
Numérique	Nombre
Real4	Nombre
Real8	Nombre
UINT1	Nombre
UINT2	Nombre
UINT4	Nombre
UINT8	Nombre
CLOB	Chaîne

Utilisation d'Amazon Kinesis Data Streams comme cible pour AWS Database Migration Service

Vous pouvez l'utiliser AWS DMS pour migrer des données vers un flux de données Amazon Kinesis. Les flux de données Amazon Kinesis font partie du service Amazon Kinesis Data Streams.

Vous pouvez utiliser des flux de données Kinesis pour collecter et traiter des flux volumineux d'enregistrements de données en temps réel.

Un flux de données Kinesis se compose de partitions. Les partitions sont des séquences d'enregistrements de données identifiées de manière unique dans un flux. Pour plus d'informations sur les partitions dans Amazon Kinesis Data Streams, consultez Partition dans le Guide du développeur Amazon Kinesis Data Streams.

AWS Database Migration Service publie des enregistrements dans un flux de données Kinesis à l'aide de JSON. Au cours de la conversion, AWS DMS sérialise chaque enregistrement de la base de données source dans une paire attribut-valeur au format JSON ou un format de message JSON_UNFORMATED. Un format de message JSON_UNFORMATED est une chaîne JSON à une seule ligne avec un délimiteur de nouvelle ligne. Il permet à Amazon Data Firehose de fournir des données Kinesis à une destination Amazon S3, puis de les interroger à l'aide de différents moteurs de requête, dont Amazon Athena.

Utilisez le mappage d'objet pour migrer vos données de n'importe quelle source de données prise en charge vers un flux cible. Avec le mappage d'objet, vous déterminez la façon de structurer les enregistrements de données dans le flux. Vous définissez également une clé de partition pour chaque table, que Kinesis Data Streams utilise pour regrouper les données dans ses partitions.

Lors de la AWS DMS création de tables sur un point de terminaison cible Kinesis Data Streams, il crée autant de tables que dans le point de terminaison de la base de données source. AWS DMS définit également plusieurs valeurs de paramètres Kinesis Data Streams. Le coût de création de la table dépend de la quantité de données et du nombre de tables à migrer.

> **Note**
>
> L'option Mode SSL de la AWS DMS console ou de l'API ne s'applique pas à certains services de streaming de données et NoSQL tels que Kinesis et DynamoDB. Ils sont sécurisés par défaut, ce qui AWS DMS montre que le paramètre du mode SSL est égal à aucun (mode SSL = aucun). Vous n'avez pas besoin de fournir de configuration supplémentaire pour que votre point de terminaison utilise le protocole SSL. Par exemple, lorsque Kinesis est utilisé comme point de terminaison cible, il est sécurisé par défaut. Tous les appels d'API adressés à Kinesis utilisent le protocole SSL, il n'est donc pas nécessaire d'ajouter une option SSL supplémentaire sur le AWS DMS terminal. Vous pouvez placer et récupérer des données en toute sécurité via des points de terminaison SSL à l'aide du protocole HTTPS, utilisé par AWS DMS par défaut lors de la connexion à un flux de données Kinesis.

Paramètres de point de terminaison Kinesis Data Streams

Lorsque vous utilisez les points de terminaison cibles de Kinesis Data Streams, vous pouvez obtenir les détails des transactions et des contrôles à l'aide de l'option `KinesisSettings` de l'API. AWS DMS

Vous pouvez définir les paramètres de connexion de l'une des manières suivantes :

- Dans la AWS DMS console, à l'aide des paramètres du point de terminaison.
- Dans la CLI, en utilisant l'`kinesis-settings`option de la CreateEndpointcommande.

Dans l'interface CLI, utilisez les paramètres de demande de l'option `kinesis-settings` suivants :

> **Note**
>
> La prise en charge des paramètres de point de terminaison `IncludeNullAndEmpty` est disponible dans AWS DMS versions 3.4.1 et ultérieures. Mais la prise en charge des autres paramètres de point de terminaison suivants pour les cibles Kinesis Data Streams est disponible dans AWS DMS.

- `MessageFormat` : format de sortie pour les enregistrements créés sur le point de terminaison. Le format du message est `JSON` (par défaut) ou `JSON_UNFORMATTED` (une seule ligne sans onglet).
- `IncludeControlDetails` : affiche des informations de contrôle détaillées pour la définition de table, la définition de colonne et les modifications de table et de colonne dans la sortie du message Kinesis. L'argument par défaut est `false`.
- `IncludeNullAndEmpty` : inclut les colonnes NULL et vides dans la cible. L'argument par défaut est `false`.
- `IncludePartitionValue` : affiche la valeur de partition dans la sortie du message Kinesis, sauf si le type de partition est `schema-table-type`. L'argument par défaut est `false`.
- `IncludeTableAlterOperations` : inclut toutes les opérations DDL (Data Definition Language) qui modifient la table dans les données de contrôle, telles que `rename-table`, `drop-table`, `add-column`, `drop-column` et `rename-column`. L'argument par défaut est `false`.
- `IncludeTransactionDetails` : fournit des informations détaillées sur les transactions à partir de la base de données source. Ces informations comprennent un horodatage de validation, une position de journal et des valeurs pour `transaction_id`, `previous_transaction_id` et

`transaction_record_id` (le décalage d'enregistrement dans une transaction). L'argument par défaut est `false`.

- `PartitionIncludeSchemaTable` : préfixe les noms de schéma et de table aux valeurs de partition, lorsque le type de partition est `primary-key-type`. Cela augmente la distribution des données entre les fragments Kinesis. Par exemple, supposons qu'un schéma `SysBench` comporte des milliers de tables et que chaque table n'ait qu'une plage limitée pour une clé primaire. Dans ce cas, la même clé primaire est envoyée à partir de milliers de tables vers le même fragment, ce qui provoque une limitation. L'argument par défaut est `false`.

L'exemple suivant illustre l'option `kinesis-settings` utilisée avec un exemple de commande `create-endpoint` émise à l'aide d' AWS CLI.

```
aws dms create-endpoint --endpoint-identifier=$target_name --engine-name kinesis --
endpoint-type target
--region us-east-1 --kinesis-settings
 ServiceAccessRoleArn=arn:aws:iam::333333333333:role/dms-kinesis-role,
StreamArn=arn:aws:kinesis:us-east-1:333333333333:stream/dms-kinesis-target-
doc,MessageFormat=json-unformatted,
IncludeControlDetails=true,IncludeTransactionDetails=true,IncludePartitionValue=true,PartitionI
IncludeTableAlterOperations=true
```

Paramètres de tâche de chargement complet multithread

Pour accélérer le transfert, AWS DMS prend en charge le chargement complet multithread vers une instance cible Kinesis Data Streams. DMS prend en charge ce traitement multithread avec des paramètres de tâche, notamment les suivants :

- `MaxFullLoadSubTasks` : utilisez cette option pour indiquer le nombre maximal de tables sources à charger en parallèle. DMS charge chaque table dans sa table cible Kinesis correspondante à l'aide d'une sous-tâche dédiée. La valeur par défaut est 8 ; la valeur maximale 49.
- `ParallelLoadThreads`— Utilisez cette option pour spécifier le nombre de threads AWS DMS utilisés pour charger chaque table dans sa table cible Kinesis. La valeur maximale pour une cible Kinesis Data Streams est de 32. Vous pouvez demander une augmentation de cette limite maximale.
- `ParallelLoadBufferSize` : utilisez cette option pour spécifier le nombre maximal d'enregistrements à stocker dans la mémoire tampon utilisée par les threads de chargement parallèles pour charger les données dans la cible Kinesis. La valeur par défaut est 50.

La valeur maximale est 1 000. Utilisez ce paramètre avec `ParallelLoadThreads`. `ParallelLoadBufferSize` est valide uniquement dans le cas de plusieurs threads.

- `ParallelLoadQueuesPerThread` : utilisez cette option pour spécifier le nombre de files d'attente auxquelles chaque thread simultané accède pour extraire les enregistrements de données des files d'attente et générer un chargement par lots pour la cible. La valeur par défaut est 1. Toutefois, pour les cibles Kinesis de différentes tailles de charge utile, la plage valide est comprise entre 5 et 512 files d'attente par thread.

Paramètres de tâche de chargement CDC multithread

Vous pouvez utiliser les paramètres de tâche afin d'améliorer les performances de la capture des données de modification (CDC) pour les points de terminaison cibles de streaming des données en temps réel, comme Kinesis et modifier le comportement de l'appel d'API `PutRecords`. Pour ce faire, vous pouvez spécifier le nombre de threads simultanés, les files d'attente par thread et le nombre d'enregistrements à stocker dans un tampon à l'aide de la tâche `ParallelApply*`. Par exemple, supposons que vous souhaitiez effectuer un chargement CDC et appliquer 128 threads en parallèle. Vous souhaitez également accéder à 64 files d'attente par thread, avec 50 enregistrements stockés par tampon.

Pour améliorer les performances du CDC, AWS DMS prend en charge les paramètres de tâche suivants :

- `ParallelApplyThreads`— Spécifie le nombre de threads simultanés AWS DMS utilisés lors d'un chargement CDC pour transférer des enregistrements de données vers un point de terminaison cible Kinesis. La valeur par défaut est zéro (0) et la valeur maximale est 32.
- `ParallelApplyBufferSize` : spécifie le nombre maximal d'enregistrements à stocker dans chaque file d'attente de mémoire tampon pour que les threads simultanés soient transférés vers un point de terminaison cible Kinesis lors d'un chargement CDC. La valeur par défaut est 100 et la valeur maximale est 1 000. Utilisez cette option lorsque `ParallelApplyThreads` spécifie plusieurs threads.
- `ParallelApplyQueuesPerThread` : spécifie le nombre de files d'attente auxquelles chaque thread accède pour extraire les enregistrements de données des files d'attente et générer un chargement par lots pour un point de terminaison Kinesis pendant la CDC. La valeur par défaut est 1 et la valeur maximale est 512.

Lorsque vous utilisez les paramètres de tâche `ParallelApply*`, la valeur par défaut `partition-key-type` est la valeur `primary-key` de la table, pas `schema-name.table-name`.

Utilisation d'une image antérieure pour afficher les valeurs d'origine des lignes CDC pour un flux de données Kinesis en tant que cible

Lorsque vous écrivez des mises à jour de CDC sur une cible de diffusion de données comme Kinesis, vous pouvez afficher les valeurs d'origine d'une ligne de base de données source avant de les modifier par une mise à jour. Pour ce faire, AWS DMS remplit une image antérieure des événements de mise à jour en fonction des données fournies par le moteur de base de données source.

Différents moteurs de base de données source fournissent différentes quantités d'informations pour une image antérieure :

- Oracle met uniquement à jour des colonnes si elles changent.
- PostgreSQL fournit uniquement des données pour les colonnes qui font partie de la clé primaire (modifiée ou non). Pour fournir des données pour toutes les colonnes (modifiées ou non), vous devez définir `REPLICA_IDENTITY` sur `FULL` au lieu de `DEFAULT`. Notez que vous devez choisir la valeur de `REPLICA_IDENTITY` avec soin pour chaque table. Si vous définissez `REPLICA_IDENTITY` sur `FULL`, toutes les valeurs de colonne sont écrites en continu dans la journalisation WAL. Cela peut entraîner des problèmes de performances ou de ressources avec les tables qui sont fréquemment mises à jour.
- MySQL fournit généralement des données pour toutes les colonnes, à l'exception des types de données BLOB et CLOB (modifiés ou non).

Pour activer avant l'imagerie pour ajouter des valeurs d'origine de la base de données source à la sortie AWS DMS , utilisez le paramètre de tâche `BeforeImageSettings` ou le paramètre `add-before-image-columns`. Ce paramètre applique une règle de transformation de colonne.

`BeforeImageSettings` ajoute un nouvel attribut JSON à chaque opération de mise à jour avec des valeurs collectées à partir du système de base de données source, comme indiqué ci-dessous.

```
"BeforeImageSettings": {
    "EnableBeforeImage": boolean,
    "FieldName": string,
    "ColumnFilter": pk-only (default) / non-lob / all (but only one)
}
```

> **Note**
>
> S'applique uniquement `BeforeImageSettings` aux AWS DMS tâches contenant un composant CDC, telles que les tâches à chargement complet plus CDC (qui migrent les données existantes et répliquent les modifications en cours), ou aux tâches CDC uniquement (qui répliquent uniquement les modifications de données). N'appliquez pas les `BeforeImageSettings` aux tâches à pleine charge uniquement.

Pour les options `BeforeImageSettings`, les conditions suivantes s'appliquent :

- Définissez l'option `EnableBeforeImage` sur `true` pour activer la génération d'image antérieure. L'argument par défaut est `false`.
- Utilisez l'option `FieldName` pour attribuer un nom au nouvel attribut JSON. Quand `EnableBeforeImage` est `true`, `FieldName` est obligatoire et ne peut pas être vide.
- L'option `ColumnFilter` spécifie une colonne à ajouter en utilisant la génération d'image antérieure. Pour ajouter uniquement des colonnes faisant partie des clés primaires de la table, utilisez la valeur par défaut, `pk-only`. Pour ajouter une colonne ayant une valeur d'image antérieure, utilisez `all`. Notez que l'image antérieure ne contient pas de colonnes avec des types de données LOB, tels que CLOB ou BLOB.

```
"BeforeImageSettings": {
    "EnableBeforeImage": true,
    "FieldName": "before-image",
    "ColumnFilter": "pk-only"
  }
```

> **Note**
>
> Les cibles Amazon S3 ne prennent pas en charge `BeforeImageSettings`. Pour les cibles S3, utilisez uniquement la règle de transformation `add-before-image-columns` à effectuer avant la génération d'images pendant la CDC.

Utilisation d'une règle de transformation d'image antérieure

Au lieu des paramètres de tâche, vous pouvez utiliser le paramètre `add-before-image-columns`, qui applique une règle de transformation de colonne. Avec ce paramètre, vous pouvez activer la génération d'image antérieure pendant la CDC sur des cibles de streaming de données comme Kinesis.

En utilisant `add-before-image-columns` dans une règle de transformation, vous pouvez exercer un contrôle plus précis sur les résultats de l'image antérieure. Les règles de transformation vous permettent d'utiliser un localisateur d'objets qui vous fournit un contrôle sur les tables sélectionnées pour la règle. En outre, vous pouvez enchaîner les règles de transformation, ce qui permet d'appliquer différentes règles à différentes tables. Vous pouvez ensuite manipuler les colonnes produites à l'aide d'autres règles.

> **Note**
>
> N'utilisez pas le paramètre `add-before-image-columns` avec le paramètre de tâche `BeforeImageSettings` dans la même tâche. N'utilisez pas les deux, pour une seule tâche.

Un type de règle `transformation` avec le paramètre `add-before-image-columns` d'une colonne doit fournir une section `before-image-def`. Vous en trouverez un exemple ci-dessous.

```
{
  "rule-type": "transformation",
  …
  "rule-target": "column",
  "rule-action": "add-before-image-columns",
  "before-image-def":{
    "column-filter": one-of  (pk-only / non-lob / all),
    "column-prefix": string,
    "column-suffix": string,
  }
}
```

La valeur de `column-prefix` est ajoutée à un nom de colonne et la valeur par défaut de `column-prefix` est `BI_`. La valeur de `column-suffix` est ajoutée au nom de la colonne et la valeur par défaut est vide. Ne définissez pas les deux `column-prefix` et `column-suffix` sur des chaînes vides.

Choisissez une valeur pour `column-filter`. Pour ajouter uniquement les colonnes qui font partie des clés primaires de la table, choisissez `pk-only` . Choisissez `non-lob` d'ajouter uniquement des colonnes qui ne sont pas de type LOB. Ou choisissez `all` d'ajouter une colonne qui a une valeur d'image antérieure.

Exemple de règle de transformation d'image antérieure

La règle de transformation de l'exemple suivant ajoute une nouvelle colonne appelée `BI_emp_no` dans la cible. Ainsi, une instruction comme `UPDATE employees SET emp_no = 3 WHERE emp_no = 1;` remplit le champ `BI_emp_no` avec 1. Lorsque vous écrivez des mises à jour de CDC sur des cibles Amazon S3, la colonne `BI_emp_no` permet de savoir quelle ligne d'origine a été mise à jour.

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "%",
        "table-name": "%"
      },
      "rule-action": "include"
    },
    {
      "rule-type": "transformation",
      "rule-id": "2",
      "rule-name": "2",
      "rule-target": "column",
      "object-locator": {
        "schema-name": "%",
        "table-name": "employees"
      },
      "rule-action": "add-before-image-columns",
      "before-image-def": {
        "column-prefix": "BI_",
        "column-suffix": "",
        "column-filter": "pk-only"
      }
    }
  ]
```

```
}
```

Pour plus d'informations sur l'utilisation de l'action de règle `add-before-image-columns`, consultez Règles et actions de transformation.

Conditions préalables à l'utilisation d'un flux de données Kinesis comme cible pour AWS Database Migration Service

Rôle IAM pour l'utilisation d'un flux de données Kinesis comme cible pour AWS Database Migration Service

Avant de configurer un flux de données Kinesis comme cible AWS DMS, assurez-vous de créer un rôle IAM. Ce rôle doit permettre d' AWS DMS assumer et d'accorder l'accès aux flux de données Kinesis vers lesquels la migration est en cours de migration. L'ensemble d'autorisations d'accès minimum est indiqué dans la stratégie IAM suivante.

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
     "Sid": "1",
     "Effect": "Allow",
     "Principal": {
        "Service": "dms.amazonaws.com"
     },
   "Action": "sts:AssumeRole"
   }
]
}
```

Le rôle que vous utilisez pour la migration vers un flux de données Kinesis doit bénéficier des autorisations suivantes.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStream",
                "kinesis:PutRecord",
                "kinesis:PutRecords"
            ],
            "Resource": "arn:aws:kinesis:region:accountID:stream/streamName"
        }
    ]
}
```

Accès à un flux de données Kinesis en tant que cible pour AWS Database Migration Service

Dans les AWS DMS versions 3.4.7 et supérieures, pour vous connecter à un point de terminaison Kinesis, vous devez effectuer l'une des opérations suivantes :

- Configurez DMS pour qu'il utilise les points de terminaison d'un VPC. Pour plus d'informations sur la configuration de DMS pour qu'il utilise les points de terminaison d'un VPC, consultez Configuration de points de terminaison de VPC en tant que points de terminaison sources et cibles AWS DMS.
- Configurez DMS pour qu'il utilise des routes publiques, c'est-à-dire qu'il rende votre instance de réplication publique. Pour plus d'informations sur les instances de réplication publiques, consultez Instances de réplication publiques et privées.

Limitations liées à l'utilisation de Kinesis Data Streams comme cible pour AWS Database Migration Service

Les limitations suivantes s'appliquent lorsque vous utilisez Kinesis Data Streams en tant que cible :

- AWS DMS publie chaque mise à jour d'un seul enregistrement de la base de données source sous la forme d'un enregistrement de données dans un flux de données Kinesis donné, quelles que soient les transactions. Toutefois, vous pouvez inclure des détails de transaction pour chaque enregistrement de données à l'aide des paramètres pertinents de l'API `KinesisSettings`.
- Le Mode LOB complet n'est pas pris en charge.
- La taille LOB maximale prise en charge est de 1 Mo.
- Kinesis Data Streams ne prend pas en charge la déduplication. Les applications qui consomment des données d'un flux doivent gérer les enregistrements en double. Pour plus d'informations,

consultez Gestion des enregistrements en double dans le Guide du développeur Amazon Kinesis Data Streams.

- AWS DMS prend en charge les deux formes suivantes pour les clés de partition :
  - `SchemaName.TableName` : une combinaison du nom du schéma et du nom de la table.
  - `${AttributeName}` : la valeur d'un des champs du fichier JSON, ou la clé primaire de la table dans la base de données source.
- Pour en savoir plus sur le chiffrement de vos données au repos dans Kinesis Data Streams, consultez Protection des données dans Kinesis Data Streams dans le Guide du développeur AWS Key Management Service .
- `BatchApply` n'est pas pris en charge pour un point de terminaison Kinesis. L'utilisation de l'application par lots (par exemple, le paramètre de tâche de métadonnées cible `BatchApplyEnabled`) pour une cible Kinesis peut entraîner une perte de données.
- Les cibles Kinesis ne sont prises en charge que pour un flux de données Kinesis dans le même AWS compte et le même Région AWS que l'instance de réplication.
- Lors de la migration depuis une source MySQL, les BeforeImage données n'incluent pas les types de données CLOB et BLOB. Pour de plus amples informations, veuillez consulter Utilisation d'une image antérieure pour afficher les valeurs d'origine des lignes CDC pour un flux de données Kinesis en tant que cible.
- AWS DMS ne prend pas en charge la migration de valeurs de type de `BigInt` données comportant plus de 16 chiffres. Pour contourner cette limitation, vous pouvez utiliser la règle de transformation suivante pour convertir la colonne `BigInt` en chaîne. Pour plus d'informations sur les règles de transformation, consultez Règles et actions de transformation.

```
{
    "rule-type": "transformation",
    "rule-id": "id",
    "rule-name": "name",
    "rule-target": "column",
    "object-locator": {
        "schema-name": "valid object-mapping rule action",
        "table-name": "",
        "column-name": ""
    },
    "rule-action": "change-data-type",
    "data-type": {
        "type": "string",
        "length": 20
```

```
    }
}
```

Utilisation du mappage d'objet pour migrer les données vers un flux de données Kinesis

AWS DMS utilise des règles de mappage de tables pour mapper les données de la source vers le flux de données Kinesis cible. Pour mapper des données vers un flux cible, vous utilisez un type de règles de mappage de tables qu'on appelle le mappage d'objet. Utilisez le mappage d'objet pour définir la façon dont les enregistrements de données de la source sont mappés aux enregistrements de données publiés dans le flux de données Kinesis.

Les flux de données Kinesis ne disposent pas d'une structure prédéfinie autre que le fait d'avoir une clé de partition. Dans une règle de mappage d'objets, les valeurs possibles de `partition-key-type` pour un enregistrement de données sont `schema-table`, `transaction-id`, `primary-key`, `constant` et `attribute-name`.

Pour créer une règle de mappage d'objet, spécifiez `rule-type` comme `object-mapping`. Cette règle spécifie le type de mappage d'objet que vous souhaitez utiliser.

La structure de la règle est la suivante.

```
{
    "rules": [
        {
            "rule-type": "object-mapping",
            "rule-id": "id",
            "rule-name": "name",
            "rule-action": "valid object-mapping rule action",
            "object-locator": {
                "schema-name": "case-sensitive schema name",
                "table-name": ""
            }
        }
    ]
}
```

AWS DMS prend actuellement en charge `map-record-to-record` et `map-record-to-document` en tant que seules valeurs valides pour le `rule-action` paramètre. Ces paramètres

affectent les valeurs qui ne sont pas exclues de la liste d'attributs `exclude-columns`. Les `map-record-to-document` valeurs `map-record-to-record` et indiquent le mode de AWS DMS gestion de ces enregistrements par défaut. Ces valeurs n'affectent en aucune façon les mappages d'attributs.

Utilisez `map-record-to-record` lors d'une migration d'une base de données relationnelle vers un flux de données Kinesis. Ce type de règle utilise la valeur `taskResourceId.schemaName.tableName` de la base de données relationnelle comme clé de partition dans le flux de données Kinesis et crée un attribut pour chaque colonne de la base de données source.

Lorsque vous utilisez `map-record-to-record`, notez ce qui suit :

- Ce paramètre n'affecte que les colonnes exclues par la liste `exclude-columns`.
- Pour chacune de ces colonnes, AWS DMS crée un attribut correspondant dans le sujet cible.
- AWS DMS crée cet attribut correspondant, que la colonne source soit utilisée ou non dans un mappage d'attributs.

Utilisez `map-record-to-document` pour placer des colonnes sources dans un document unique à plat du flux cible approprié en utilisant le nom d'attribut « _doc ». AWS DMS place les données dans un mappage unique à plat sur la source appelée « `_doc` ». Ce placement s'applique à toute colonne de la table source non listée dans la liste d'attributs `exclude-columns`.

Une manière de comprendre `map-record-to-record` est de le voir en action. Dans cet exemple, imaginons que vous commencez avec une ligne de table d'une base de données relationnelle, présentant la structure et les données suivantes :

FirstName	LastName	StoreId	HomeAddress	HomePhone	WorkAddress	WorkPhone	DateofBirth
Randy	Marsh	5	221B Baker Street	1234567890	31 Spooner Street, Quahog	9876543210	02/29/1988

Pour migrer ces informations d'un schéma nommé `Test` vers un flux de données Kinesis, vous créez des règles pour mapper les données au flux cible. La règle suivante illustre ce mappage.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "rule-action": "include",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            }
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "2",
            "rule-name": "DefaultMapToKinesis",
            "rule-action": "map-record-to-record",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "Customers"
            }
        }
    ]
}
```

L'exemple qui suit illustre le format d'enregistrement résultant dans le flux de données Kinesis :

- StreamName: XXX
- PartitionKey: Test.Customers //SchmaName.TableName
- Data : //The following JSON message

```
  {
     "FirstName": "Randy",
     "LastName": "Marsh",
     "StoreId":  "5",
     "HomeAddress": "221B Baker Street",
     "HomePhone": "1234567890",
     "WorkAddress": "31 Spooner Street, Quahog",
     "WorkPhone": "9876543210",
     "DateOfBirth": "02/29/1988"
  }
```

Supposons toutefois que vous utilisiez les mêmes règles, mais que vous redéfinissiez le paramètre `rule-action` sur `map-record-to-document` et que vous excluiez certaines colonnes. La règle suivante illustre ce mappage.

```
{
 "rules": [
    {
   "rule-type": "selection",
   "rule-id": "1",
   "rule-name": "1",
   "rule-action": "include",
   "object-locator": {
    "schema-name": "Test",
    "table-name": "%"
   }
  },
  {
   "rule-type": "object-mapping",
   "rule-id": "2",
   "rule-name": "DefaultMapToKinesis",
   "rule-action": "map-record-to-document",
   "object-locator": {
    "schema-name": "Test",
    "table-name": "Customers"
   },
   "mapping-parameters": {
    "exclude-columns": [
     "homeaddress",
     "homephone",
     "workaddress",
     "workphone"
    ]
   }
  }
 ]
}
```

Dans ce cas, les colonnes non répertoriées dans le paramètre `exclude-columns`, `FirstName`, `LastName`, `StoreId` et `DateOfBirth`, sont mappées à `_doc`. L'exemple qui suit illustre le format d'enregistrement résultant.

```
      {
           "data":{
               "_doc":{
                   "FirstName": "Randy",
                   "LastName": "Marsh",
                   "StoreId":  "5",
                   "DateOfBirth": "02/29/1988"
               }
           }
       }
```

Restructuration de données avec le mappage d'attribut

Vous pouvez restructurer les données lors de leur migration vers un flux de données Kinesis à l'aide d'un mappage d'attribut. Par exemple, vous pourriez vouloir regrouper plusieurs champs de la source en un seul champ dans la cible. Le mappage d'attribut suivant illustre comment restructurer les données.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "rule-action": "include",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            }
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "2",
            "rule-name": "TransformToKinesis",
            "rule-action": "map-record-to-record",
            "target-table-name": "CustomerData",
```

```
            "object-locator": {
                "schema-name": "Test",
                "table-name": "Customers"
            },
            "mapping-parameters": {
                "partition-key-type": "attribute-name",
                "partition-key-name": "CustomerName",
                "exclude-columns": [
                    "firstname",
                    "lastname",
                    "homeaddress",
                    "homephone",
                    "workaddress",
                    "workphone"
                ],
                "attribute-mappings": [
                    {
                        "target-attribute-name": "CustomerName",
                        "attribute-type": "scalar",
                        "attribute-sub-type": "string",
                        "value": "${lastname}, ${firstname}"
                    },
                    {
                        "target-attribute-name": "ContactDetails",
                        "attribute-type": "document",
                        "attribute-sub-type": "json",
                        "value": {
                            "Home": {
                                "Address": "${homeaddress}",
                                "Phone": "${homephone}"
                            },
                            "Work": {
                                "Address": "${workaddress}",
                                "Phone": "${workphone}"
                            }
                        }
                    }
                ]
            }
        }
    ]
}
```

Pour définir une valeur constante pour `partition-key`, spécifiez une valeur `partition-key`. Par exemple, vous pouvez le faire pour forcer le stockage de toutes les données dans une seule partition. Le mappage suivant illustre cette approche.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "1",
            "rule-name": "TransformToKinesis",
            "rule-action": "map-record-to-document",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "Customer"
            },
            "mapping-parameters": {
                "partition-key": {
                    "value": "ConstantPartitionKey"
                },
                "exclude-columns": [
                    "FirstName",
                    "LastName",
                    "HomeAddress",
                    "HomePhone",
                    "WorkAddress",
                    "WorkPhone"
                ],
                "attribute-mappings": [
                    {
                        "attribute-name": "CustomerName",
                        "value": "${FirstName},${LastName}"
                    },
                    {
```

```
                        "attribute-name": "ContactDetails",
                        "value": {
                            "Home": {
                                "Address": "${HomeAddress}",
                                "Phone": "${HomePhone}"
                            },
                            "Work": {
                                "Address": "${WorkAddress}",
                                "Phone": "${WorkPhone}"
                            }
                        }
                    },
                    {
                        "attribute-name": "DateOfBirth",
                        "value": "${DateOfBirth}"
                    }
                ]
            }
        }
    ]
}
```

> **Note**
>
> La valeur `partition-key` d'un enregistrement de contrôle correspondant à une table spécifique est `TaskId.SchemaName.TableName`. La valeur `partition-key` d'un enregistrement de contrôle correspondant à une tâche spécifique est le `TaskId` de cet enregistrement. La spécification d'une valeur `partition-key` dans le mappage d'objet n'a aucun impact sur la `partition-key` d'un enregistrement de contrôle.

Format de message pour Kinesis Data Streams

La sortie JSON est simplement une liste de paires clé-valeur. Un format de message JSON_UNFORMATED est une chaîne JSON à une seule ligne avec un délimiteur de nouvelle ligne.

AWS DMS fournit les champs réservés suivants pour faciliter la consommation des données issues des Kinesis Data Streams :

RecordType

Les enregistrements peuvent être de type Données ou Contrôle. Les enregistrements de données représentent les lignes réelles de la source. Les enregistrements de contrôle sont destinés à des événements importants dans le flux, par exemple un redémarrage de la tâche.

Opération

Pour les enregistrements de données, l'opération peut être `load`,`insert`, `update` ou `delete`.

Pour les enregistrements de contrôle, l'opération peut être `create-table`, `rename-table`, `drop-table`, `change-columns`, `add-column`, `drop-column`, `rename-column` ou `column-type-change`.

SchemaName

Schéma source de l'enregistrement. Ce champ peut être vide pour un enregistrement de contrôle.

TableName

Table source de l'enregistrement. Ce champ peut être vide pour un enregistrement de contrôle.

Horodatage

Horodatage de la construction du message JSON. Le champ est formaté selon le format ISO 8601.

Utiliser Apache Kafka comme cible pour AWS Database Migration Service

Vous pouvez l'utiliser AWS DMS pour migrer des données vers un cluster Apache Kafka. Apache Kafka est une plateforme de streaming distribuée. Apache Kafka vous permet d'ingérer et de traiter des données de streaming en temps réel.

AWS propose également Amazon Managed Streaming pour Apache Kafka (Amazon MSK) à utiliser comme cible. AWS DMS Amazon MSK est un service de streaming Apache Kafka entièrement géré qui simplifie l'implémentation et la gestion des instances Apache Kafka. Il fonctionne avec les versions open source d'Apache Kafka, et vous accédez aux instances Amazon MSK en tant que AWS DMS cibles, exactement comme n'importe quelle instance Apache Kafka. Pour plus d'informations, consultez Qu'est-ce qu'Amazon MSK ? dans le Guide du développeur Amazon Managed Streaming for Apache Kafka.

Un cluster Kafka stocke les flux d'enregistrements dans des catégories appelées « rubriques », divisées en partitions. Les partitions sont des séquences identifiées de manière unique

d'enregistrements de données (messages) dans une rubrique. Les partitions peuvent être réparties entre plusieurs agents dans un cluster pour permettre le traitement parallèle des enregistrements d'une rubrique. Pour de plus amples informations sur les rubriques et les partitions et leur distribution dans Apache Kafka, veuillez consulter Rubriques et journaux et distribution.

Votre cluster Kafka peut être une instance Amazon MSK, un cluster exécuté sur une instance Amazon EC2 ou un cluster sur site. Une instance Amazon MSK ou un cluster sur une instance Amazon EC2 peut se trouver dans le même VPC ou dans un autre VPC. Dans le cas d'un cluster sur site, vous pouvez utiliser votre propre serveur de noms sur site pour votre instance de réplication afin de résoudre le nom d'hôte du cluster. Pour en savoir plus sur la configuration d'un serveur de noms pour votre instance de réplication, consultez Utilisation de votre propre serveur de noms sur site. Pour plus d'informations sur la configuration d'un réseau, consultez Configuration d'un réseau pour une instance de réplication.

Lorsque vous utilisez un cluster Amazon MSK, assurez-vous que son groupe de sécurité autorise l'accès à partir de votre instance de réplication. Pour en savoir plus sur la modification du groupe de sécurité d'un cluster Amazon MSK, consultez Modification du groupe de sécurité d'un cluster Amazon MSK.

AWS Database Migration Service publie des enregistrements dans un sujet Kafka à l'aide de JSON. Au cours de la conversion, AWS DMS sérialise chaque enregistrement de la base de données source dans une paire attribut-valeur au format JSON.

Utilisez le mappage d'objet pour migrer vos données de n'importe quelle source de données prise en charge vers un cluster Kafka cible. Avec le mappage d'objet, vous déterminez la façon de structurer les enregistrements de données dans la rubrique cible. Vous définissez également une clé de partition pour chaque table, qu'Apache Kafka utilise pour regrouper les données dans ses partitions.

Actuellement, ne AWS DMS prend en charge qu'un seul sujet par tâche. Pour une seule tâche comportant plusieurs tables, tous les messages sont placés dans une seule rubrique. Chaque message inclut une section de métadonnées qui identifie le schéma et la table cibles. AWS DMS les versions 3.4.6 et supérieures prennent en charge la réplication multisujet à l'aide du mappage d'objets. Pour de plus amples informations, veuillez consulter Réplication à plusieurs rubriques à l'aide du mappage d'objet.

Paramètres du point de terminaison Apache Kafka

Vous pouvez spécifier les détails de connexion via les paramètres du point de terminaison dans la AWS DMS console ou via l'`--kafka-settings`option de la CLI. Les conditions requises pour chaque paramètre sont les suivantes :

- `Broker` : spécifiez les emplacements d'un ou de plusieurs agents dans votre cluster Kafka sous la forme d'une liste séparée par des virgules de tous les éléments *`broker-hostname`*`:`*`port`*. Par exemple : `"ec2-12-345-678-901.compute-1.amazonaws.com:2345,ec2-10-987-654-321.compute-1.a` Ce paramètre peut spécifier les emplacements d'un ou de tous les agents du cluster. Les agents de cluster communiquent tous pour gérer le partitionnement des enregistrements de données migrés vers la rubrique.
- `Topic` : (facultatif) spécifiez le nom de rubrique avec une longueur maximale de 255 lettres et symboles. Vous pouvez utiliser le point (.), le trait de soulignement (_) et le moins (-). Les noms de rubrique avec un point (.) ou un trait de soulignement (_) peuvent entrer en collision dans des structures de données internes. Utilisez l'un ou l'autre de ces symboles, mais pas les deux dans le nom de la rubrique. Si vous ne spécifiez pas de nom de rubrique, AWS DMS `"kafka-default-topic"` utilisez-le comme rubrique de migration.

  > **Note**
  >
  > Pour AWS DMS créer soit un sujet de migration que vous spécifiez, soit le sujet par défaut, défini dans le `auto.create.topics.enable = true` cadre de la configuration de votre cluster Kafka. Pour plus d'informations, consultez Limitations liées à l'utilisation d'Apache Kafka comme cible pour AWS Database Migration Service.

- `MessageFormat` : format de sortie pour les enregistrements créés sur le point de terminaison. Le format du message est `JSON` (par défaut) ou `JSON_UNFORMATTED` (une seule ligne sans onglet).
- `MessageMaxBytes` : taille maximale en octets des enregistrements créés sur le point de terminaison. La valeur par défaut est 1 000 000.

  > **Note**
  >
  > Vous ne pouvez utiliser la AWS CLI/SDK que pour passer à une valeur autre que celle par `MessageMaxBytes` défaut. Par exemple, pour modifier votre point de terminaison Kafka existant et redéfinir la valeur de `MessageMaxBytes`, utilisez la commande suivante.
  >
  > ```
  > aws dms modify-endpoint --endpoint-arn your-endpoint
  > --kafka-settings Broker="broker1-server:broker1-port,broker2-server:broker2-
  > port,...",
  > Topic=topic-name,MessageMaxBytes=integer-of-max-message-size-in-bytes
  > ```

- `IncludeTransactionDetails` : fournit des informations détaillées sur les transactions à partir de la base de données source. Ces informations comprennent un horodatage de validation, une position de journal et des valeurs pour `transaction_id`, `previous_transaction_id` et `transaction_record_id`(le décalage d'enregistrement dans une transaction). L'argument par défaut est `false`.
- `IncludePartitionValue` : affiche la valeur de partition dans la sortie du message Kafka, sauf si le type de partition est `schema-table-type`. L'argument par défaut est `false`.
- `PartitionIncludeSchemaTable` : préfixe les noms de schéma et de table aux valeurs de partition, lorsque le type de partition est `primary-key-type`. Cela augmente la distribution des données entre les partitions Kafka. Par exemple, supposons qu'un schéma `SysBench` comporte des milliers de tables et que chaque table n'ait qu'une plage limitée pour une clé primaire. Dans ce cas, la même clé primaire est envoyée à partir de milliers de tables vers la même partition, ce qui provoque une limitation. L'argument par défaut est `false`.
- `IncludeTableAlterOperations` : inclut toutes les opérations DDL (Data Definition Language) qui modifient la table dans les données de contrôle, telles que `rename-table`, `drop-table`, `add-column`, `drop-column` et `rename-column`. L'argument par défaut est `false`.
- `IncludeControlDetails` : affiche les informations de contrôle détaillées pour la définition de table, la définition de colonne et les modifications de table et de colonne dans la sortie du message Kafka. L'argument par défaut est `false`.
- `IncludeNullAndEmpty` : inclut les colonnes NULL et vides dans la cible. L'argument par défaut est `false`.
- `SecurityProtocol` : définit une connexion sécurisée à un point de terminaison cible Kafka à l'aide du protocole TLS (Transport Layer Security). Les options sont `ssl-authentication`, `ssl-encryption` et `sasl-ssl`. L'utilisation de `sasl-ssl` requiert `SaslUsername` et `SaslPassword`.
- `SslEndpointIdentificationAlgorithm`— Définit la vérification du nom d'hôte pour le certificat. Ce paramètre est pris en charge dans les AWS DMS versions 3.5.1 et ultérieures. Les options disponibles sont les suivantes :
  - `NONE`: désactive la vérification du nom d'hôte du broker dans la connexion client.
  - `HTTPS`: Activez la vérification du nom d'hôte du broker dans la connexion client.

Vous pouvez augmenter la vitesse du transfert dans les paramètres. Pour ce faire, AWS DMS prend en charge un chargement complet multithread sur un cluster cible Apache Kafka. AWS DMS prend en charge ce multithreading avec des paramètres de tâche qui incluent les éléments suivants :

- `MaxFullLoadSubTasks`— Utilisez cette option pour indiquer le nombre maximum de tables sources à charger en parallèle. AWS DMS charge chaque table dans la table cible Kafka correspondante à l'aide d'une sous-tâche dédiée. La valeur par défaut est 8 ; la valeur maximale 49.
- `ParallelLoadThreads`— Utilisez cette option pour spécifier le nombre de threads AWS DMS utilisés pour charger chaque table dans sa table cible Kafka. La valeur maximale pour une cible Apache Kafka est 32. Vous pouvez demander une augmentation de cette limite maximale.
- `ParallelLoadBufferSize` : utilisez cette option pour spécifier le nombre maximal d'enregistrements à stocker dans la mémoire tampon utilisée par les threads de chargement parallèles pour charger les données dans la cible Kafka. La valeur par défaut est 50. La valeur maximale est 1 000. Utilisez ce paramètre avec `ParallelLoadThreads`. `ParallelLoadBufferSize` est valide uniquement dans le cas de plusieurs threads.
- `ParallelLoadQueuesPerThread` : utilisez cette option pour spécifier le nombre de files d'attente auxquelles chaque thread simultané accède pour extraire les enregistrements de données des files d'attente et générer un chargement par lots pour la cible. La valeur par défaut est 1. La valeur maximale est 512.

Vous pouvez améliorer les performances de capture des données de modification (CDC) pour les points de terminaison Kafka en ajustant les paramètres de tâche des threads parallèles et des opérations en bloc. Pour ce faire, vous pouvez spécifier le nombre de threads simultanés, les files d'attente par thread et le nombre d'enregistrements à stocker dans un tampon à l'aide de la tâche `ParallelApply*`. Par exemple, supposons que vous souhaitiez effectuer un chargement CDC et appliquer 128 threads en parallèle. Vous souhaitez également accéder à 64 files d'attente par thread, avec 50 enregistrements stockés par tampon.

Pour améliorer les performances du CDC, AWS DMS prend en charge les paramètres de tâche suivants :

- `ParallelApplyThreads`— Spécifie le nombre de threads simultanés AWS DMS utilisés lors d'un chargement CDC pour transférer des enregistrements de données vers un point de terminaison cible Kafka. La valeur par défaut est zéro (0) et la valeur maximale est 32.
- `ParallelApplyBufferSize` : spécifie le nombre maximal d'enregistrements à stocker dans chaque file d'attente de mémoire tampon pour que les threads simultanés soient transférés vers un point de terminaison cible Kafka lors d'un chargement CDC. La valeur par défaut est 100 et la valeur maximale est 1 000. Utilisez cette option lorsque `ParallelApplyThreads` spécifie plusieurs threads.

- `ParallelApplyQueuesPerThread` : spécifie le nombre de files d'attente auxquelles chaque thread accède pour extraire les enregistrements de données des files d'attente et générer un chargement par lots pour un point de terminaison Kafka pendant la CDC. La valeur par défaut est 1. La valeur maximale est 512.

Lorsque vous utilisez les paramètres de tâche `ParallelApply*`, la valeur par défaut `partition-key-type` est la valeur `primary-key` de la table, pas `schema-name.table-name`.

Connexion à Kafka à l'aide du protocole TLS (Transport Layer Security)

Un cluster Kafka accepte les connexions sécurisées à l'aide du protocole TLS (Transport Layer Security). Avec DMS, vous pouvez utiliser l'une des trois options de protocole de sécurité suivantes pour sécuriser la connexion d'un point de terminaison Kafka.

Chiffrement SSL (**`server-encryption`**)

Les clients valident l'identité du serveur par le biais du certificat du serveur. Une connexion chiffrée est alors établie entre le serveur et le client.

Authentification SSL (**`mutual-authentication`**)

Le serveur et le client valident mutuellement leur identité respective via leurs propres certificats. Une connexion chiffrée est alors établie entre le serveur et le client.

SASL-SSL (**`mutual-authentication`**)

La méthode SASL (Simple Authentication and Security Layer) remplace le certificat du client par un nom d'utilisateur et un mot de passe pour valider l'identité du client. Plus précisément, vous fournissez un nom d'utilisateur et un mot de passe enregistrés par le serveur afin que ce dernier puisse valider l'identité d'un client. Une connexion chiffrée est alors établie entre le serveur et le client.

Important

Apache Kafka et Amazon MSK acceptent les certificats résolus. Il s'agit d'une limitation connue de Kafka et d'Amazon MSK qui doit être corrigée. Pour plus d'informations, consultez Apache Kafka issues, KAFKA-3700.
Si vous utilisez Amazon MSK, envisagez d'utiliser des listes de contrôle d'accès (ACL) pour contourner cette limitation connue. Pour plus d'informations sur l'utilisation des listes

ACL, consultez la section Listes de contrôle d'accès (ACL) Apache Kafka dans le Guide du développeur Amazon Managed Streaming for Apache Kafka.
Si vous utilisez un cluster Kafka autogéré, consultez Comment dated 21/Oct/18 pour en savoir plus sur la configuration de votre cluster.

Utilisation du chiffrement SSL avec Amazon MSK ou un cluster Kafka autogéré

Vous pouvez utiliser le chiffrement SSL pour sécuriser la connexion d'un point de terminaison à Amazon MSK ou à un cluster Kafka autogéré. Lorsque vous utilisez la méthode d'authentification par chiffrement SSL, les clients valident l'identité d'un serveur par le biais du certificat du serveur. Une connexion chiffrée est alors établie entre le serveur et le client.

Pour utiliser le chiffrement SSL pour vous connecter à Amazon MSK

- Définissez le paramètre de point de terminaison du protocole de sécurité (`SecurityProtocol`) à l'aide de l'option `ssl-encryption` lorsque vous créez votre point de terminaison Kafka cible.

  L'exemple de code JSON suivant définit le protocole de sécurité sur le chiffrement SSL.

```
"KafkaSettings": {
    "SecurityProtocol": "ssl-encryption",
}
```

Pour utiliser le chiffrement SSL pour un cluster Kafka autogéré

1. Si vous utilisez une autorité de certification (CA) privée dans votre cluster Kafka sur site, chargez votre certificat de CA privée et obtenez un Amazon Resource Name (ARN).
2. Définissez le paramètre de point de terminaison du protocole de sécurité (`SecurityProtocol`) à l'aide de l'option `ssl-encryption` lorsque vous créez votre point de terminaison Kafka cible. L'exemple de code JSON suivant définit le protocole de sécurité sur `ssl-encryption`.

```
"KafkaSettings": {
    "SecurityProtocol": "ssl-encryption",
}
```

3. Si vous utilisez une CA privée, définissez `SslCaCertificateArn` dans l'ARN obtenu à la première étape ci-dessus.

Utilisation de l'authentification SSL

Vous pouvez utiliser l'authentification SSL pour sécuriser la connexion d'un point de terminaison à Amazon MSK ou à un cluster Kafka autogéré.

Pour activer l'authentification client et le chiffrement à l'aide de l'authentification SSL pour se connecter à Amazon MSK, procédez comme suit :

- Préparez une clé privée et un certificat public pour Kafka.
- Chargez les certificats dans le gestionnaire de certificats DMS.
- Créez un point de terminaison cible Kafka avec les ARN de certificat correspondants spécifiés dans les paramètres de point de terminaison Kafka.

Pour préparer une clé privée et un certificat public pour Amazon MSK

1. Créez une instance EC2 et configurez un client de sorte qu'il utilise l'authentification, comme décrit dans les étapes 1 à 9 de la section Authentification TLS mutuelle du Guide du développeur Amazon Managed Streaming for Apache Kafka.

   Après avoir effectué ces étapes, vous disposez d'un Certificate-ARN (l'ARN de certificat public enregistré dans ACM) et d'une clé privée contenus dans un fichier `kafka.client.keystore.jks`.

2. Obtenez le certificat public et copiez-le dans le fichier `signed-certificate-from-acm.pem` à l'aide de la commande suivante :

   ```
   aws acm-pca get-certificate --certificate-authority-arn Private_CA_ARN --
   certificate-arn Certificate_ARN
   ```

   Cette commande renvoie des informations semblables à celles de l'exemple suivant :

   ```
   {"Certificate": "123", "CertificateChain": "456"}
   ```

Vous copiez ensuite votre équivalent de `"123"` dans le fichier `signed-certificate-from-acm.pem`.

3. Pour obtenir la clé privée, importez la clé `msk-rsa` à partir de `kafka.client.keystore.jks to keystore.p12`, comme illustré dans l'exemple suivant.

```
keytool -importkeystore \
-srckeystore kafka.client.keystore.jks \
-destkeystore keystore.p12 \
-deststoretype PKCS12 \
-srcalias msk-rsa-client \
-deststorepass test1234 \
-destkeypass test1234
```

4. Utilisez la commande suivante pour exporter `keystore.p12` au format `.pem`.

```
Openssl pkcs12 -in keystore.p12 -out encrypted-private-client-key.pem –nocerts
```

Le message Entrer une phrase passe PEM apparaît et identifie la clé appliquée pour chiffrer le certificat.

5. Supprimez les attributs de conteneur et les attributs de clé du fichier `.pem` pour vous assurer que la première ligne commence par la chaîne suivante.

```
                              ---BEGIN ENCRYPTED PRIVATE KEY---
```

Pour charger un certificat public et une clé privée dans le gestionnaire de certificats DMS et tester la connexion à Amazon MSK

1. Chargez dans le gestionnaire de certificats DMS à l'aide de la commande suivante.

```
aws dms import-certificate --certificate-identifier signed-cert --certificate-pem
 file://path to signed cert
aws dms import-certificate --certificate-identifier private-key –certificate-pem
 file://path to private key
```

2. Créez un point de terminaison cible Amazon MSK et testez la connexion pour vous assurer que l'authentification TLS fonctionne.

```
aws dms create-endpoint --endpoint-identifier $endpoint-identifier --engine-name
 kafka --endpoint-type target --kafka-settings
'{"Broker": "b-0.kafka260.aaaaa1.a99.kafka.us-east-1.amazonaws.com:0000",
 "SecurityProtocol":"ssl-authentication",
"SslClientCertificateArn": "arn:aws:dms:us-east-1:012346789012:cert:",
"SslClientKeyArn": "arn:aws:dms:us-
east-1:0123456789012:cert:","SslClientKeyPassword":"test1234"}'
aws dms test-connection -replication-instance-arn=$rep_inst_arn –endpoint-arn=
$kafka_tar_arn_msk
```

Important

Vous pouvez utiliser l'authentification SSL pour sécuriser une connexion à un cluster Kafka autogéré. Dans certains cas, vous devrez peut-être utiliser une autorité de certification (CA) privée dans votre cluster Kafka sur site. Si tel est le cas, chargez votre chaîne de CA, votre certificat public et votre clé privée dans le gestionnaire de certificats DMS. Utilisez ensuite l'Amazon Resource Name (ARN) correspondant dans vos paramètres de point de terminaison lorsque vous créez votre point de terminaison cible Kafka sur site.

Pour préparer une clé privée et un certificat signé pour un cluster Kafka autogéré

1. Générez une paire de clés comme indiqué dans l'exemple suivant.

```
keytool -genkey -keystore kafka.server.keystore.jks -validity 300 -storepass your-
keystore-password
-keypass your-key-passphrase -dname "CN=your-cn-name"
-alias alias-of-key-pair -storetype pkcs12 -keyalg RSA
```

2. Générez une demande de signature de certificat (CSR).

```
keytool -keystore kafka.server.keystore.jks -certreq -file server-cert-sign-
request-rsa -alias on-premise-rsa -storepass your-key-store-password
-keypass your-key-password
```

3. Utilisez la CA du magasin de clés de confiance de votre cluster pour signer la CSR. Si vous n'avez pas de CA, vous pouvez créer votre propre CA privée.

```
openssl req -new -x509 -keyout ca-key -out ca-cert -days validate-days
```

4. Importez `ca-cert` dans le magasin de clés de confiance et le magasin de clés du serveur. Si vous n'avez pas de magasin de clés de confiance, utilisez la commande suivante pour le créer et y importer `ca-cert` .

```
keytool -keystore kafka.server.truststore.jks -alias CARoot -import -file ca-cert
keytool -keystore kafka.server.keystore.jks -alias CARoot -import -file ca-cert
```

5. Signez le certificat.

```
openssl x509 -req -CA ca-cert -CAkey ca-key -in server-cert-sign-request-rsa -out
 signed-server-certificate.pem
-days validate-days -CAcreateserial -passin pass:ca-password
```

6. Importez le certificat signé dans le magasin de clés.

```
keytool -keystore kafka.server.keystore.jks -import -file signed-certificate.pem -
alias on-premise-rsa -storepass your-keystore-password
-keypass your-key-password
```

7. Utilisez la commande suivante pour importer la clé `on-premise-rsa` de `kafka.server.keystore.jks` dans `keystore.p12`.

```
keytool -importkeystore \
-srckeystore kafka.server.keystore.jks \
-destkeystore keystore.p12 \
-deststoretype PKCS12 \
-srcalias on-premise-rsa \
-deststorepass your-truststore-password \
-destkeypass your-key-password
```

8. Utilisez la commande suivante pour exporter `keystore.p12` au format `.pem`.

```
Openssl pkcs12 -in keystore.p12 -out encrypted-private-server-key.pem –nocerts
```

9. Chargez `encrypted-private-server-key.pem`, `signed-certificate.pem` et `ca-cert` dans le gestionnaire de certificats DMS.
10. Créez un point de terminaison en utilisant les ARN renvoyés.

```
aws dms create-endpoint --endpoint-identifier $endpoint-identifier --engine-name
 kafka --endpoint-type target --kafka-settings
'{"Broker": "b-0.kafka260.aaaaa1.a99.kafka.us-east-1.amazonaws.com:9092",
 "SecurityProtocol":"ssl-authentication",
"SslClientCertificateArn": "your-client-cert-arn","SslClientKeyArn": "your-client-
key-arn","SslClientKeyPassword":"your-client-key-password",
"SslCaCertificateArn": "your-ca-certificate-arn"}'

aws dms test-connection -replication-instance-arn=$rep_inst_arn –endpoint-arn=
$kafka_tar_arn_msk
```

Utilisation de l'authentification SASL-SSL pour se connecter à Amazon MSK

La méthode SASL (Simple Authentication and Security Layer) utilise un nom d'utilisateur et un mot de passe pour valider l'identité d'un client et établit une connexion chiffrée entre le serveur et le client.

Pour utiliser SASL, vous devez commencer par créer un nom d'utilisateur et un mot de passe sécurisés lorsque vous configurez votre cluster Amazon MSK. Pour savoir comment configurer un nom d'utilisateur et un mot de passe sécurisés pour un cluster Amazon MSK, consultez Configuration de l'authentification SASL/SCRAM pour un cluster Amazon MSK dans le Guide du développeur Amazon Managed Streaming for Apache Kafka.

Ensuite, lorsque vous créez votre point de terminaison cible Kafka, définissez le paramètre de point de terminaison du protocole de sécurité (`SecurityProtocol`) à l'aide de l'option `sasl-ssl`. Vous définissez également les options `SaslUsername` et `SaslPassword`. Assurez-vous qu'elles sont cohérentes avec le nom d'utilisateur et le mot de passe sécurisés que vous avez créés lorsque vous avez configuré votre cluster Amazon MSK pour la première fois, comme illustré dans l'exemple JSON suivant.

```
"KafkaSettings": {
    "SecurityProtocol": "sasl-ssl",
    "SaslUsername":"Amazon MSK cluster secure user name",
    "SaslPassword":"Amazon MSK cluster secure password"
}
```

> **Note**
>
> - Actuellement, ne AWS DMS prend en charge que le protocole SASL-SSL public soutenu par une autorité de certification. DMS ne prend pas en charge le protocole SASL-SSL pour une utilisation avec un cluster Kafka autogéré basé sur une CA privée.
> - Pour l'authentification SASL-SSL, AWS DMS prend en charge le mécanisme SCRAM-SHA-512 par défaut. AWS DMS les versions 3.5.0 et supérieures prennent également en charge le mécanisme Plain. Pour prendre en charge le mécanisme Plain, définissez le paramètre `SaslMechanism` du type de données d'API `KafkaSettings` sur `PLAIN`.

Utilisation d'une image antérieure pour afficher les valeurs originales des lignes de la CDC pour Apache Kafka comme cible

Lorsque vous écrivez des mises à jour de CDC sur une cible de diffusion de données comme Kafka, vous pouvez afficher les valeurs d'origine d'une ligne de base de données source avant de les modifier par une mise à jour. Pour ce faire, AWS DMS remplit une image antérieure des événements de mise à jour en fonction des données fournies par le moteur de base de données source.

Différents moteurs de base de données source fournissent différentes quantités d'informations pour une image antérieure :

- Oracle met uniquement à jour des colonnes si elles changent.
- PostgreSQL fournit uniquement des données pour les colonnes qui font partie de la clé primaire (modifiée ou non). Si la réplication logique est utilisée et que REPLICA IDENTITY FULL est défini pour la table source, vous pouvez obtenir les informations antérieures et postérieures complètes sur la ligne écrite dans les journaux WAL et disponible ici.
- MySQL fournit généralement des données pour toutes les colonnes (modifiées ou non).

Pour activer avant l'imagerie pour ajouter des valeurs d'origine de la base de données source à la sortie AWS DMS , utilisez le paramètre de tâche `BeforeImageSettings` ou le paramètre `add-before-image-columns`. Ce paramètre applique une règle de transformation de colonne.

`BeforeImageSettings` ajoute un nouvel attribut JSON à chaque opération de mise à jour avec des valeurs collectées à partir du système de base de données source, comme indiqué ci-dessous.

```
"BeforeImageSettings": {
    "EnableBeforeImage": boolean,
    "FieldName": string,
    "ColumnFilter": pk-only (default) / non-lob / all (but only one)
}
```

> **Note**
>
> Appliquez les `BeforeImageSettings` à la charge complète et aux tâches CDC (qui migrent les données existantes et répliquent les modifications en cours), ou aux tâches CDC uniquement (qui répliquent les modifications de données uniquement). N'appliquez pas les `BeforeImageSettings` aux tâches à pleine charge uniquement.

Pour les options `BeforeImageSettings`, les conditions suivantes s'appliquent :

- Définissez l'option `EnableBeforeImage` sur `true` pour activer la génération d'image antérieure. L'argument par défaut est `false`.
- Utilisez l'option `FieldName` pour attribuer un nom au nouvel attribut JSON. Quand `EnableBeforeImage` est `true`, `FieldName` est obligatoire et ne peut pas être vide.
- L'option `ColumnFilter` spécifie une colonne à ajouter en utilisant la génération d'image antérieure. Pour ajouter uniquement des colonnes faisant partie des clés primaires de la table, utilisez la valeur par défaut, `pk-only`. Pour ajouter uniquement des colonnes qui ne sont pas de type LOB, utilisez `non-lob`. Pour ajouter une colonne ayant une valeur d'image antérieure, utilisez `all`.

  ```
  "BeforeImageSettings": {
      "EnableBeforeImage": true,
      "FieldName": "before-image",
      "ColumnFilter": "pk-only"
  ```

```
}
```

Utilisation d'une règle de transformation d'image antérieure

Au lieu des paramètres de tâche, vous pouvez utiliser le paramètre `add-before-image-columns`, qui applique une règle de transformation de colonne. Avec ce paramètre, vous pouvez activer la génération d'image antérieure pendant la CDC sur des cibles de diffusion de données telles que Kafka.

En utilisant `add-before-image-columns` dans une règle de transformation, vous pouvez exercer un contrôle plus précis sur les résultats de l'image antérieure. Les règles de transformation vous permettent d'utiliser un localisateur d'objets qui vous fournit un contrôle sur les tables sélectionnées pour la règle. En outre, vous pouvez enchaîner les règles de transformation, ce qui permet d'appliquer différentes règles à différentes tables. Vous pouvez ensuite manipuler les colonnes produites à l'aide d'autres règles.

> **Note**
>
> N'utilisez pas le paramètre `add-before-image-columns` avec le paramètre de tâche `BeforeImageSettings` dans la même tâche. N'utilisez pas les deux, pour une seule tâche.

Un type de règle `transformation` avec le paramètre `add-before-image-columns` d'une colonne doit fournir une section `before-image-def`. Vous en trouverez un exemple ci-dessous.

```
{
  "rule-type": "transformation",
  …
  "rule-target": "column",
  "rule-action": "add-before-image-columns",
  "before-image-def":{
    "column-filter": one-of  (pk-only / non-lob / all),
    "column-prefix": string,
    "column-suffix": string,
  }
}
```

La valeur de `column-prefix` est ajoutée à un nom de colonne et la valeur par défaut de `column-prefix` est `BI_`. La valeur de `column-suffix` est ajoutée au nom de la colonne et la valeur par

défaut est vide. Ne définissez pas les deux `column-prefix` et `column-suffix` sur des chaînes vides.

Choisissez une valeur pour `column-filter`. Pour ajouter uniquement les colonnes qui font partie des clés primaires de la table, choisissez `pk-only` . Choisissez `non-lob` d'ajouter uniquement des colonnes qui ne sont pas de type LOB. Ou choisissez `all` d'ajouter une colonne qui a une valeur d'image antérieure.

Exemple de règle de transformation d'image antérieure

La règle de transformation de l'exemple suivant ajoute une nouvelle colonne appelée `BI_emp_no` dans la cible. Ainsi, une instruction comme `UPDATE employees SET emp_no = 3 WHERE emp_no = 1;` remplit le champ `BI_emp_no` avec 1. Lorsque vous écrivez des mises à jour de CDC sur des cibles Amazon S3, la colonne `BI_emp_no` permet de savoir quelle ligne d'origine a été mise à jour.

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "%",
        "table-name": "%"
      },
      "rule-action": "include"
    },
    {
      "rule-type": "transformation",
      "rule-id": "2",
      "rule-name": "2",
      "rule-target": "column",
      "object-locator": {
        "schema-name": "%",
        "table-name": "employees"
      },
      "rule-action": "add-before-image-columns",
      "before-image-def": {
        "column-prefix": "BI_",
        "column-suffix": "",
        "column-filter": "pk-only"
```

```
      }
    }
  ]
}
```

Pour plus d'informations sur l'utilisation de l'action de règle `add-before-image-columns`, consultez Règles et actions de transformation.

Limitations liées à l'utilisation d'Apache Kafka comme cible pour AWS Database Migration Service

Les limitations suivantes s'appliquent lorsque vous utilisez apache Kafka comme cible :

- AWS DMS Les points de terminaison cibles de Kafka ne prennent pas en charge le contrôle d'accès IAM pour Amazon Managed Streaming for Apache Kafka (Amazon MSK).
- Le Mode LOB complet n'est pas pris en charge.
- Spécifiez un fichier de configuration Kafka pour votre cluster avec des propriétés permettant AWS DMS de créer automatiquement de nouveaux sujets. Incluez le paramètre, `auto.create.topics.enable = true`. Si vous utilisez Amazon MSK, vous pouvez spécifier la configuration par défaut lorsque vous créez votre cluster Kafka, puis redéfinir le paramètre `auto.create.topics.enable` sur `true`. Pour plus d'informations sur les paramètres de configuration par défaut, consultez Configuration Amazon MSK par défaut dans le Guide du développeur Amazon Managed Streaming for Apache Kafka. Si vous devez modifier un cluster Kafka existant créé à l'aide d'Amazon MSK, exécutez la AWS CLI commande `aws kafka create-configuration` pour mettre à jour votre configuration Kafka, comme dans l'exemple suivant :

```
14:38:41 $ aws kafka create-configuration --name "kafka-configuration" --kafka-
versions "2.2.1" --server-properties file://~/kafka_configuration
{
    "LatestRevision": {
        "Revision": 1,
        "CreationTime": "2019-09-06T14:39:37.708Z"
    },
    "CreationTime": "2019-09-06T14:39:37.708Z",
    "Name": "kafka-configuration",
    "Arn": "arn:aws:kafka:us-east-1:111122223333:configuration/kafka-
configuration/7e008070-6a08-445f-9fe5-36ccf630ecfd-3"
}
```

Ici, `//~/kafka_configuration` est le fichier de configuration que vous avez créé avec les paramètres de propriété requis.

Si vous utilisez votre propre instance Kafka installée sur Amazon EC2, modifiez la configuration du cluster Kafka avec `auto.create.topics.enable = true` le paramètre AWS DMS permettant de créer automatiquement de nouveaux sujets, en utilisant les options fournies avec votre instance.

- AWS DMS publie chaque mise à jour d'un seul enregistrement de la base de données source sous la forme d'un enregistrement de données (message) dans un sujet Kafka donné, quelles que soient les transactions.
- AWS DMS prend en charge les deux formes suivantes pour les clés de partition :
  - `SchemaName.TableName` : une combinaison du nom du schéma et du nom de la table.
  - `${AttributeName}` : la valeur d'un des champs du fichier JSON, ou la clé primaire de la table dans la base de données source.
- `BatchApply` n'est pas pris en charge pour un point de terminaison Kafka. L'utilisation de l'application par lots (par exemple, le paramètre de tâche de métadonnées cible `BatchApplyEnabled`) pour une cible Kafka peut entraîner une perte de données.
- AWS DMS ne prend pas en charge la migration de valeurs de type de `BigInt` données comportant plus de 16 chiffres. Pour contourner cette limitation, vous pouvez utiliser la règle de transformation suivante pour convertir la colonne `BigInt` en chaîne. Pour plus d'informations sur les règles de transformation, consultez Règles et actions de transformation.

```
{
    "rule-type": "transformation",
    "rule-id": "id",
    "rule-name": "name",
    "rule-target": "column",
    "object-locator": {
        "schema-name": "valid object-mapping rule action",
        "table-name": "",
        "column-name": ""
    },
    "rule-action": "change-data-type",
    "data-type": {
        "type": "string",
        "length": 20
    }
}
```

Utilisation du mappage d'objet pour migrer les données vers une rubrique Kafka

AWS DMS utilise des règles de mappage de tables pour mapper les données de la source vers le sujet Kafka cible. Pour mapper des données à une rubrique cible, vous utilisez un type de règle de mappage de table appelé « mappage d'objet ». Vous utilisez le mappage d'objet pour définir la façon dont les enregistrements de données de la source sont mappés sur les enregistrements de données publiés dans une rubrique Kafka.

Les rubriques Kafka ne disposent pas d'une structure prédéfinie autre que le fait d'avoir une clé de partition.

> **Note**
>
> Vous n'avez pas besoin d'utiliser le mappage d'objet. Vous pouvez utiliser un mappage de table standard pour différentes transformations. Cependant, le type de clé de partition suivra les comportements par défaut suivants :
>
> - La clé primaire est utilisée comme clé de partition pour le chargement complet.
> - Si aucun paramètre de tâche parallel-apply n'est utilisé, `schema.table` est utilisé comme clé de partition pour la CDC.
> - Si des paramètres de tâche parallel-apply sont utilisés, la clé primaire est utilisée comme clé de partition pour la CDC.

Pour créer une règle de mappage d'objet, spécifiez `rule-type` comme `object-mapping`. Cette règle spécifie le type de mappage d'objet que vous souhaitez utiliser.

La structure de la règle est la suivante.

```
{
    "rules": [
        {
            "rule-type": "object-mapping",
            "rule-id": "id",
            "rule-name": "name",
            "rule-action": "valid object-mapping rule action",
            "object-locator": {
                "schema-name": "case-sensitive schema name",
                "table-name": ""
            }
```

```
        }
    ]
}
```

AWS DMS prend actuellement en charge `map-record-to-record` et `map-record-to-document` en tant que seules valeurs valides pour le `rule-action` paramètre. Ces paramètres affectent les valeurs qui ne sont pas exclues de la liste d'attributs `exclude-columns`. Les `map-record-to-document` valeurs `map-record-to-record` et indiquent le mode de AWS DMS gestion de ces enregistrements par défaut. Ces valeurs n'affectent en aucune façon les mappages d'attributs.

Utilisez `map-record-to-record` lors d'une migration d'une base de données relationnelle vers une rubrique Kafka. Ce type de règle utilise la valeur `taskResourceId.schemaName.tableName` de la base de données relationnelle comme clé de partition dans la rubrique Kafka, et crée un attribut pour chaque colonne dans la base de données source.

Lorsque vous utilisez `map-record-to-record`, notez ce qui suit :

- Ce paramètre n'affecte que les colonnes exclues par la liste `exclude-columns`.
- Pour chacune de ces colonnes, AWS DMS crée un attribut correspondant dans le sujet cible.
- AWS DMS crée cet attribut correspondant, que la colonne source soit utilisée ou non dans un mappage d'attributs.

Une manière de comprendre `map-record-to-record` est de le voir en action. Dans cet exemple, imaginons que vous commencez avec une ligne de table d'une base de données relationnelle, présentant la structure et les données suivantes :

FirstName	LastName	StoreId	HomeAddre ss	HomePhon	WorkAddre ss	WorkPhone	DateofBir th
Randy	Marsh	5	221B Baker Street	123456789 0	31 Spooner Street, Quahog	987654321 0	02/29/198 8

Pour migrer ces informations à partir d'un schéma nommé `Test` vers une rubrique Kafka, vous créez des règles pour mapper les données sur la rubrique cible. La règle suivante illustre ce mappage.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "rule-action": "include",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            }
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "2",
            "rule-name": "DefaultMapToKafka",
            "rule-action": "map-record-to-record",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "Customers"
            }
        }
    ]
}
```

Compte tenu d'une rubrique Kafka et d'une clé de partition (dans ce cas, `taskResourceId.schemaName.tableName`), l'exemple ci-dessous illustre le format d'enregistrement résultant à l'aide de nos exemples de données dans la rubrique cible Kafka :

```
  {
     "FirstName": "Randy",
     "LastName": "Marsh",
     "StoreId":  "5",
     "HomeAddress": "221B Baker Street",
     "HomePhone": "1234567890",
     "WorkAddress": "31 Spooner Street, Quahog",
     "WorkPhone": "9876543210",
     "DateOfBirth": "02/29/1988"
  }
```

Rubriques

- Restructuration de données avec le mappage d'attribut
- Réplication à plusieurs rubriques à l'aide du mappage d'objet
- Format de message pour Apache Kafka

Restructuration de données avec le mappage d'attribut

Vous pouvez restructurer les données lors de leur migration vers une rubrique Kafka à l'aide d'un mappage d'attribut. Par exemple, vous pourriez vouloir regrouper plusieurs champs de la source en un seul champ dans la cible. Le mappage d'attribut suivant illustre comment restructurer les données.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "rule-action": "include",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            }
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "2",
            "rule-name": "TransformToKafka",
            "rule-action": "map-record-to-record",
            "target-table-name": "CustomerData",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "Customers"
            },
            "mapping-parameters": {
                "partition-key-type": "attribute-name",
                "partition-key-name": "CustomerName",
                "exclude-columns": [
                    "firstname",
                    "lastname",
                    "homeaddress",
                    "homephone",
                    "workaddress",
                    "workphone"
```

```
                ],
                "attribute-mappings": [
                    {
                        "target-attribute-name": "CustomerName",
                        "attribute-type": "scalar",
                        "attribute-sub-type": "string",
                        "value": "${lastname}, ${firstname}"
                    },
                    {
                        "target-attribute-name": "ContactDetails",
                        "attribute-type": "document",
                        "attribute-sub-type": "json",
                        "value": {
                            "Home": {
                                "Address": "${homeaddress}",
                                "Phone": "${homephone}"
                            },
                            "Work": {
                                "Address": "${workaddress}",
                                "Phone": "${workphone}"
                            }
                        }
                    }
                ]
            }
        }
    ]
}
```

Pour définir une valeur constante pour `partition-key`, spécifiez une valeur `partition-key`. Par exemple, vous pouvez le faire pour forcer le stockage de toutes les données dans une seule partition. Le mappage suivant illustre cette approche.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            },
```

```
            "rule-action": "include"
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "1",
            "rule-name": "TransformToKafka",
            "rule-action": "map-record-to-document",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "Customer"
            },
            "mapping-parameters": {
                "partition-key": {
                    "value": "ConstantPartitionKey"
                },
                "exclude-columns": [
                    "FirstName",
                    "LastName",
                    "HomeAddress",
                    "HomePhone",
                    "WorkAddress",
                    "WorkPhone"
                ],
                "attribute-mappings": [
                    {
                        "attribute-name": "CustomerName",
                        "value": "${FirstName},${LastName}"
                    },
                    {
                        "attribute-name": "ContactDetails",
                        "value": {
                            "Home": {
                                "Address": "${HomeAddress}",
                                "Phone": "${HomePhone}"
                            },
                            "Work": {
                                "Address": "${WorkAddress}",
                                "Phone": "${WorkPhone}"
                            }
                        }
                    },
                    {
                        "attribute-name": "DateOfBirth",
                        "value": "${DateOfBirth}"
```

```
                    }
                ]
            }
        }
    ]
}
```

> **Note**
>
> La valeur `partition-key` d'un enregistrement de contrôle correspondant à une table spécifique est `TaskId.SchemaName.TableName`. La valeur `partition-key` d'un enregistrement de contrôle correspondant à une tâche spécifique est le `TaskId` de cet enregistrement. La spécification d'une valeur `partition-key` dans le mappage d'objet n'a aucun impact sur la `partition-key` d'un enregistrement de contrôle.

Réplication à plusieurs rubriques à l'aide du mappage d'objet

Par défaut, les AWS DMS tâches migrent toutes les données sources vers l'une des rubriques Kafka suivantes :

- Comme indiqué dans le champ Rubrique du point de terminaison AWS DMS cible.
- Comme indiqué par `kafka-default-topic` si le champ Rubrique du point de terminaison cible n'est pas renseigné et que le paramètre `auto.create.topics.enable` Kafka est défini sur `true`.

Avec les versions 3.4.6 et supérieures AWS DMS du moteur, vous pouvez utiliser l'`kafka-target-topic`attribut pour associer chaque table source migrée à une rubrique distincte. Par exemple, les règles de mappage d'objet suivantes migrent les tables sources `Customer` et `Address` vers les rubriques Kafka `customer_topic` et `address_topic`, respectivement. Dans le même temps, AWS DMS migre toutes les autres tables sources, y compris la `Bills` table du `Test` schéma, vers le sujet spécifié dans le point de terminaison cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
```

```
            "rule-action": "include",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            }
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "2",
            "rule-name": "MapToKafka1",
            "rule-action": "map-record-to-record",
            "kafka-target-topic": "customer_topic",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "Customer"
            },
            "partition-key": {"value": "ConstantPartitionKey" }
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "3",
            "rule-name": "MapToKafka2",
            "rule-action": "map-record-to-record",
            "kafka-target-topic": "address_topic",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "Address"
            },
            "partition-key": {"value": "HomeAddress" }
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "4",
            "rule-name": "DefaultMapToKafka",
            "rule-action": "map-record-to-record",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "Bills"
            }
        }
    ]
}
```

En utilisant la réplication à plusieurs rubriques Kafka, vous pouvez regrouper et migrer les tables sources afin de séparer les rubriques Kafka à l'aide d'une seule tâche de réplication.

Format de message pour Apache Kafka

La sortie JSON est simplement une liste de paires clé-valeur.

RecordType

Les enregistrements peuvent être de type Données ou Contrôle. Les enregistrements de données représentent les lignes réelles de la source. Les enregistrements de contrôle sont destinés à des événements importants dans le flux, par exemple un redémarrage de la tâche.

Opération

Pour les enregistrements de données, l'opération peut être `load`,`insert`, `update` ou `delete`.

Pour les enregistrements de contrôle, l'opération peut être `create-table`, `rename-table`, `drop-table`, `change-columns`, `add-column`, `drop-column`, `rename-column` ou `column-type-change`.

SchemaName

Schéma source de l'enregistrement. Ce champ peut être vide pour un enregistrement de contrôle.

TableName

Table source de l'enregistrement. Ce champ peut être vide pour un enregistrement de contrôle.

Horodatage

Horodatage de la construction du message JSON. Le champ est formaté selon le format ISO 8601.

L'exemple de message JSON suivant illustre un message de type de données avec toutes les métadonnées supplémentaires.

```
{
   "data":{
      "id":100000161,
      "fname":"val61s",
      "lname":"val61s",
      "REGION":"val61s"
   },
```

```
   "metadata":{
      "timestamp":"2019-10-31T22:53:59.721201Z",
      "record-type":"data",
      "operation":"insert",
      "partition-key-type":"primary-key",
      "partition-key-value":"sbtest.sbtest_x.100000161",
      "schema-name":"sbtest",
      "table-name":"sbtest_x",
      "transaction-id":9324410911751,
      "transaction-record-id":1,
      "prev-transaction-id":9324410910341,
      "prev-transaction-record-id":10,
      "commit-timestamp":"2019-10-31T22:53:55.000000Z",
      "stream-position":"mysql-bin-
changelog.002171:36912271:0:36912333:9324410911751:mysql-bin-changelog.002171:36912209"
   }
}
```

L'exemple de message JSON suivant illustre un message de type contrôle.

```
{
   "control":{
      "table-def":{
         "columns":{
            "id":{
               "type":"WSTRING",
               "length":512,
               "nullable":false
            },
            "fname":{
               "type":"WSTRING",
               "length":255,
               "nullable":true
            },
            "lname":{
               "type":"WSTRING",
               "length":255,
               "nullable":true
            },
            "REGION":{
               "type":"WSTRING",
               "length":1000,
```

```
               "nullable":true
            }
         },
         "primary-key":[
            "id"
         ],
         "collation-name":"latin1_swedish_ci"
      }
   },
   "metadata":{
      "timestamp":"2019-11-21T19:14:22.223792Z",
      "record-type":"control",
      "operation":"create-table",
      "partition-key-type":"task-id",
      "schema-name":"sbtest",
      "table-name":"sbtest_t1"
   }
}
```

Utilisation d'un cluster Amazon OpenSearch Service en tant que cible pour AWS Database Migration Service

Vous pouvez utiliser AWS DMS pour migrer des données vers Amazon OpenSearch Service (OpenSearch Service). OpenSearch Service est un service géré qui facilite le déploiement, l'utilisation et la mise à l'échelle d'un cluster OpenSearch Service.

Dans OpenSearch Service, vous utilisez des index et des documents. Un index est un ensemble de documents et un document est un objet JSON contenant des valeurs scalaires, des tableaux et d'autres objets. OpenSearch fournit un langage de requête basé sur JSON afin que vous puissiez interroger les données d'un index et récupérer les documents correspondants.

Lorsque AWS DMS crée des index pour un point de terminaison cible pour OpenSearch Service, il crée un index pour chaque table du point de terminaison source. Le coût de la création d'un index OpenSearch Service dépend de plusieurs facteurs. Il s'agit du nombre d'index créés, du volume total de données dans ces index et de la petite quantité de métadonnées qu'OpenSearch stocke pour chaque document.

Configurez votre cluster OpenSearch Service avec des ressources de calcul et de stockage appropriées selon la portée de votre migration. Nous vous recommandons de prendre en compte les facteurs suivants, selon la tâche de réplication que vous souhaitez utiliser :

- Pour un chargement de données complet, tenez compte du volume total de données à migrer ainsi que de la vitesse du transfert.
- Pour la réplication de modifications en cours, tenez compte de la fréquence des mises à jour et de vos exigences en termes de latence.

En outre, configurez les paramètres d'index sur votre cluster OpenSearch, en prêtant une attention particulière au nombre de documents.

Paramètres de tâche de chargement complet multithread

Pour augmenter la vitesse du transfert, AWS DMS prend en charge un chargement complet multithread vers un cluster cible OpenSearch Service. AWS DMS prend en charge ce multithreading avec les paramètres de tâche suivants :

- `MaxFullLoadSubTasks` : utilisez cette option pour indiquer le nombre maximal de tables sources à charger en parallèle. DMS charge chaque table dans son index cible OpenSearch Service correspondant à l'aide d'une sous-tâche dédiée. La valeur par défaut est 8 ; la valeur maximale 49.
- `ParallelLoadThreads` : utilisez cette option pour spécifier le nombre de threads utilisés par AWS DMS pour charger chaque table dans son index cible OpenSearch Service. La valeur maximale pour une cible OpenSearch Service est 32. Vous pouvez demander une augmentation de cette limite maximale.

  > **Note**
  >
  > Si vous ne modifiez pas la valeur par défaut de `ParallelLoadThreads` (0), AWS DMS transfère un seul enregistrement à la fois. Cette approche entraîne une charge inutile sur votre cluster OpenSearch Service. Veillez à définir cette option sur 1 ou plus.

- `ParallelLoadBufferSize` : utilisez cette option pour spécifier le nombre maximal d'enregistrements à stocker dans la mémoire tampon utilisée par les threads de chargement parallèles pour charger les données dans la cible OpenSearch Service. La valeur par défaut est 50. La valeur maximale est 1 000. Utilisez ce paramètre avec `ParallelLoadThreads`. `ParallelLoadBufferSize` est valide uniquement dans le cas de plusieurs threads.

Pour plus d'informations sur la façon dont DMS charge un cluster OpenSearch Service à l'aide du multithreading, consultez le billet de blog AWS Mise à l'échelle d'Amazon OpenSearch Service pour les migrations AWS Database Migration Service (langue française non garantie).

Paramètres de tâche de chargement CDC multithread

Vous pouvez utiliser les paramètres de tâche afin d'améliorer les performances de la capture des données de modification (CDC) pour un cluster cible OpenSearch Service et modifier le comportement de l'appel d'API `PutRecords`. Pour ce faire, vous pouvez spécifier le nombre de threads simultanés, les files d'attente par thread et le nombre d'enregistrements à stocker dans un tampon à l'aide de la tâche `ParallelApply*`. Par exemple, supposons que vous souhaitiez effectuer un chargement CDC et appliquer 32 threads en parallèle. Vous souhaitez également accéder à 64 files d'attente par thread, avec 50 enregistrements stockés par tampon.

> **Note**
>
> La prise en charge de l'utilisation des paramètres de tâche `ParallelApply*` pendant la CDC vers des points de terminaison cibles Amazon OpenSearch Service est disponible dans AWS DMS versions 3.4.0 et ultérieures.

Pour accroître les performances de la fonctionnalité CDC, AWS DMS prend en charge les paramètres de tâche suivants :

- `ParallelApplyThreads` : spécifie le nombre de threads simultanés utilisés par AWS DMS pendant un chargement CDC pour transférer des enregistrements de données vers un point de terminaison cible OpenSearch Service. La valeur par défaut est zéro (0) et la valeur maximale est 32.
- `ParallelApplyBufferSize` : spécifie le nombre maximal d'enregistrements à stocker dans chaque file d'attente de mémoire tampon pour que les threads simultanés soient transférés vers un point de terminaison cible OpenSearch Service lors d'un chargement CDC. La valeur par défaut est 100 et la valeur maximale est 1 000. Utilisez cette option lorsque `ParallelApplyThreads` spécifie plusieurs threads.
- `ParallelApplyQueuesPerThread` : spécifie le nombre de files d'attente auxquelles chaque thread accède pour extraire les enregistrements de données des files d'attente et générer un chargement par lots pour un point de terminaison OpenSearch Service pendant la CDC.

Lorsque vous utilisez les paramètres de tâche `ParallelApply*`, la valeur par défaut `partition-key-type` est la valeur `primary-key` de la table, pas `schema-name.table-name`.

Migration d'une table de base de données relationnelle vers un index OpenSearch Service

AWS DMS prend en charge la migration de données vers les types de données scalaires d'OpenSearch Service. Lors d'une migration depuis une base de données relationnelle telle qu'Oracle ou MySQL vers OpenSearch Service, vous pouvez restructurer la manière dont vous stockez les données.

AWS DMS prend en charge les types de données scalaires OpenSearch Service suivants :

- Booléen
- Date
- Float
- Int
- Chaîne

AWS DMS convertit les données de type Date en type String. Vous pouvez spécifier des mappages personnalisés pour interpréter ces dates.

AWS DMS ne prend pas en charge la migration de types de données LOB.

Conditions préalables à l'utilisation d'Amazon OpenSearch Service en tant que cible pour AWS Database Migration Service

Avant de commencer à utiliser une base de données OpenSearch Service en tant que cible pour AWS DMS, veillez à créer un rôle AWS Identity and Access Management (IAM). Ce rôle doit permettre à AWS DMS d'accéder aux index OpenSearch Service au point de terminaison cible. L'ensemble d'autorisations d'accès minimum est indiqué dans la stratégie IAM suivante.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Principal": {
                "Service": "dms.amazonaws.com"
```

```
        },
        "Action": "sts:AssumeRole"
        }
    ]
}
```

Le rôle que vous utilisez pour la migration vers OpenSearch Service doit bénéficier des autorisations suivantes.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                      "es:ESHttpDelete",
                      "es:ESHttpGet",
                      "es:ESHttpHead",
                      "es:ESHttpPost",
                      "es:ESHttpPut"
                    ],
            "Resource": "arn:aws:es:region:account-id:domain/domain-name/*"
        }
    ]
}
```

Dans l'exemple précédent, remplacez *region* par l'identifiant de région AWS, *account-id* par votre ID de compte AWS et *domain-name* par le nom de votre domaine Amazon OpenSearch Service. Voici un exemple : `arn:aws:es:us-west-2:123456789012:domain/my-es-domain`.

Paramètres de point de terminaison lors de l'utilisation d'OpenSearch Service en tant que cible pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données cible OpenSearch Service comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison cible à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--elasticsearch-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec OpenSearch Service en tant que cible sont indiqués dans le tableau suivant.

Nom d'attribut	Valeurs valides	Valeur par défaut et description
`FullLoadErrorPercentage`	Un nombre entier positif supérieur à 0, mais ne dépassant pas 100.	10 - Pour une tâche de chargement complet, cet attribut détermine le seuil d'erreurs autorisées avant que la tâche échoue. Par exemple, supposons qu'il y a 1 500 lignes au point de terminaison source et que ce paramètre est défini sur 10. Dans ce cas, la tâche échoue si AWS DMS rencontre plus de 150 erreurs (10 % du nombre de lignes) lors de l'écriture sur le point de terminaison cible.
`ErrorRetryDuration`	Un nombre entier positif supérieur à 0.	300 - Si une erreur se produit au point de terminaison cible, AWS DMS effectue de nouvelles tentatives pendant ce nombre de secondes. Sinon, la tâche échoue.

Limitations liées à l'utilisation d'Amazon OpenSearch Service en tant que cible pour AWS Database Migration Service

Les limitations suivantes s'appliquent à l'utilisation d'Amazon OpenSearch Service en tant que cible :

- OpenSearch Service utilise le mappage dynamique (auto guess) pour déterminer les types de données à utiliser pour les données migrées.
- OpenSearch Service stocke chaque document avec un ID unique. Voici un exemple d'ID.

  ```
  "_id": "D359F8B537F1888BC71FE20B3D79EAE6674BE7ACA9B645B0279C7015F6FF19FD"
  ```

  Chaque ID de document compte 64 octets. Tenez-en compte dans les exigences de stockage. Par exemple, si vous migrez 100 000 lignes à partir d'une source AWS DMS, l'index OpenSearch Service généré nécessite le stockage de 6 400 000 octets supplémentaires.
- Avec OpenSearch Service, vous ne pouvez pas mettre à jour les attributs de clé primaire. Cette restriction est importante lors de l'utilisation de la réplication continue avec la capture des données modifiées (CDC), car elle peut entraîner des données non souhaitées dans la cible. Dans le mode

CDC, les clés primaires sont mappées vers des valeurs SHA256, dont la longueur est de 32 octets. Ces valeurs sont converties en chaînes de 64 octets lisibles par l'homme et sont utilisées comme ID de document OpenSearch Service.

- Si AWS DMS détecte des éléments qui ne peuvent pas être migrés, il consigne les messages d'erreur dans Amazon CloudWatch Logs. Ce comportement diffère de celui des autres points de terminaison cibles de AWS DMS, qui enregistrent les erreurs dans une table d'exceptions.
- AWS DMS ne prend pas en charge la connexion à un cluster Amazon ES sur lequel le contrôle d'accès détaillé est activé avec un utilisateur principal et un mot de passe.
- AWS DMS ne prend pas en charge OpenSearch Service sans serveur.
- OpenSearch Service ne prend pas en charge l'écriture de données dans des index préexistants.

Types de données cibles pour Amazon OpenSearch Service

Lorsque AWS DMS migre des données à partir de bases de données hétérogènes, le service mappe les types de données de la base de données source vers des types de données intermédiaires, appelés types de données AWS DMS. Le service mappe ensuite les types de données intermédiaires vers les types de données cibles. Le tableau suivant illustre chaque type de données AWS DMS et le type de données auquel il est mappé dans OpenSearch Service.

Type de données AWS DMS	Type de données OpenSearch Service
Booléen	boolean
Date	chaîne
Heure	date
Horodatage	date
INT4	entier
Real4	float
UINT4	entier

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Utilisation d'Amazon DocumentDB en tant que cible pour AWS Database Migration Service

Pour en savoir plus sur les versions d'Amazon DocumentDB (compatible avec MongoDB) prises en charge par AWS DMS, consultez Objectifs pour AWS DMS. Vous pouvez utiliser AWS DMS pour migrer des données vers Amazon DocumentDB (compatible avec MongoDB) à partir de tout moteur de données source pris en charge par AWS DMS. Le moteur source peut se trouver sur un service géré par AWS tel qu'Amazon RDS, Aurora ou Amazon S3. Le moteur peut également se trouver sur une base de données autogérée, telle que MongoDB, qui s'exécute sur Amazon EC2 ou sur site.

Vous pouvez utiliser AWS DMS pour répliquer des données source dans les bases de données, les collections ou les documents Amazon DocumentDB.

> **Note**
>
> Si votre point de terminaison source est MongoDB ou Amazon DocumentDB, exécutez la migration en mode document.

MongoDB stocke les données au format JSON binaire (BSON). AWS DMS prend en charge tous les types de données BSON qui sont pris en charge par Amazon DocumentDB. Pour obtenir la liste de ces types de données, consultez API, opérations et types de données MongoDB pris en charge dans le Guide du développeur Amazon DocumentDB.

Si le point de terminaison source est une base de données relationnelle, AWS DMS mappe les objets de base de données à Amazon DocumentDB comme suit :

- Une base de données relationnelle, ou schéma de base de données, est mappée à une base de données Amazon DocumentDB.
- Les tables situées dans une base de données relationnelle sont mappées à des collections dans Amazon DocumentDB.
- Les enregistrements situés dans une table relationnelle sont mappés à des documents dans Amazon DocumentDB. Chaque document est construit à partir de données de l'enregistrement source.

Si le point de terminaison source est Amazon S3, les objets Amazon DocumentDB résultants correspondent aux règles de mappage AWS DMS pour Amazon S3. Prenons l'exemple de l'URI suivant :

```
s3://mybucket/hr/employee
```

Dans ce cas, AWS DMS mappe les objets de `mybucket` à Amazon DocumentDB comme suit :

- La partie supérieure de l'URI (`hr`) est mappée à une base de données Amazon DocumentDB.
- La partie suivante de l'URI (`employee`) est mappée à une collection Amazon DocumentDB.
- Chaque objet de `employee` est mappé à un document dans Amazon DocumentDB.

Pour plus d'informations sur les règles de mappage pour Amazon S3, consultez Utilisation d'Amazon S3 en tant que source pour AWS DMS.

Paramètres de point de terminaison Amazon DocumentDB

Dans AWS DMS versions 3.5.0 et ultérieures, vous pouvez améliorer les performances de capture des données de modification (CDC) pour les points de terminaison Amazon DocumentDB en ajustant les paramètres de tâche des threads parallèles et des opérations en bloc. Pour ce faire, vous pouvez spécifier le nombre de threads simultanés, les files d'attente par thread et le nombre d'enregistrements à stocker dans un tampon à l'aide de la tâche `ParallelApply*`. Par exemple, supposons que vous souhaitiez effectuer un chargement CDC et appliquer 128 threads en parallèle. Vous souhaitez également accéder à 64 files d'attente par thread, avec 50 enregistrements stockés par tampon.

Pour accroître les performances de la fonctionnalité CDC, AWS DMS prend en charge les paramètres de tâche suivants :

- `ParallelApplyThreads` : spécifie le nombre de threads simultanés utilisés par AWS DMS pendant un chargement CDC pour transférer des enregistrements de données vers un point de terminaison cible Amazon DocumentDB. La valeur par défaut est zéro (0) et la valeur maximale est 32.
- `ParallelApplyBufferSize` : spécifie le nombre maximal d'enregistrements à stocker dans chaque file d'attente de mémoire tampon pour que les threads simultanés soient transférés vers un point de terminaison cible Amazon DocumentDB lors d'un chargement CDC. La valeur par défaut est 100 et la valeur maximale est 1 000. Utilisez cette option lorsque `ParallelApplyThreads` spécifie plusieurs threads.

- `ParallelApplyQueuesPerThread` : spécifie le nombre de files d'attente auxquelles chaque thread accède pour extraire les enregistrements de données des files d'attente et générer un chargement par lots pour un point de terminaison Amazon DocumentDB pendant la CDC. La valeur par défaut est 1. La valeur maximale est 512.

Pour plus de détails sur l'utilisation d'Amazon DocumentDB en tant que cible pour AWS DMS, consultez les sections suivantes :

Rubriques

- Mappage de données d'une source vers une cible Amazon DocumentDB
- Connexion aux clusters Elastic Amazon DocumentDB en tant que cible
- Réplication continue avec Amazon DocumentDB en tant que cible
- Limitations de l'utilisation d'Amazon DocumentDB en tant que cible
- Utilisation des paramètres de point de terminaison avec Amazon DocumentDB en tant que cible
- Types de données cibles pour Amazon DocumentDB

> **Note**
>
> Pour une présentation détaillée du processus de migration, consultez Migration de MongoDB vers Amazon DocumentDB dans le guide de migration étape par étape d'AWS Database Migration Service.

Mappage de données d'une source vers une cible Amazon DocumentDB

AWS DMS lit les enregistrements du point de terminaison source et crée des documents JSON en fonction des données lues. Pour chaque document JSON, AWS DMS doit déterminer un champ `_id` faisant office d'identifiant unique. Il écrit ensuite le document JSON dans une collection Amazon DocumentDB en utilisant le champ `_id` comme clé primaire.

Données source se composant d'une colonne unique

Si les données source se composent d'une seule colonne, elles doivent être de type string (chaîne). (En fonction du moteur source, le type de données réel peut être VARCHAR, NVARCHAR, TEXT, LOB, CLOB, etc.). AWS DMS suppose que les données sont un document JSON valide et les réplique telles quelles sur Amazon DocumentDB.

Si le document JSON résultant contient un champ nommé `_id`, ce dernier est utilisé comme `_id` unique dans Amazon DocumentDB.

Si le document JSON ne contient pas de champ `_id`, Amazon DocumentDB génère une valeur `_id` automatiquement.

Données source se composant de plusieurs colonnes

Si les données source se composent de plusieurs colonnes, AWS DMS crée un document JSON à partir de l'ensemble de ces colonnes. Pour déterminer le champ `_id` du document, AWS DMS procède comme suit :

- Si l'une des colonnes est nommée `_id`, les données de cette colonne sont utilisées comme `_id` cible.
- S'il n'y a pas de colonne `_id`, mais que les données source comportent une clé primaire ou un index unique, AWS DMS utilise cette valeur de clé ou d'index comme valeur `_id`. Les données de la clé primaire ou de l'index unique apparaissent également sous forme de champs explicites dans le document JSON.
- S'il n'y a pas de colonne `_id` et aucune clé primaire ni aucun index unique, Amazon DocumentDB génère une valeur `_id` automatiquement.

Imposition d'un type de données au niveau du point de terminaison cible

AWS DMS peut modifier les structures de données lors de l'écriture sur un point de terminaison cible Amazon DocumentDB. Vous pouvez demander ces modifications en renommant des colonnes et des tables au niveau du point de terminaison source ou en fournissant des règles de transformation qui sont appliquées lorsqu'une tâche est en cours d'exécution.

Utilisation d'un document JSON imbriqué (préfixe json_)

Pour imposer un type de données, vous pouvez préfixer le nom de colonne source avec `json_` (c'est-à-dire, `json_`*`columnName`*), manuellement ou à l'aide d'une transformation. Dans ce cas, la colonne est créée sous la forme d'un document JSON imbriqué dans le document cible, plutôt que sous la forme d'un champ de chaîne.

Par exemple, supposons que vous souhaitiez migrer le document suivant à partir d'un point de terminaison source MongoDB.

```
{
    "_id": "1",
```

```
    "FirstName": "John",
    "LastName": "Doe",
    "ContactDetails": "{"Home": {"Address": "Boston","Phone": "1111111"},"Work":
 { "Address": "Boston", "Phone": "2222222222"}}"
}
```

Si vous n'imposez aucun des types de données source, le document `ContactDetails` imbriqué est migré sous la forme d'une chaîne.

```
{
    "_id": "1",
    "FirstName": "John",
    "LastName": "Doe",
    "ContactDetails": "{\"Home\": {\"Address\": \"Boston\",\"Phone\": \"1111111\"},
\"Work\": { \"Address\": \"Boston\", \"Phone\": \"2222222222\"}}"
}
```

Toutefois, vous pouvez ajouter une règle de transformation pour contraindre l'insertion de `ContactDetails` dans un objet JSON. Par exemple, supposons que le nom de la colonne source d'origine est `ContactDetails`. Pour forcer à appliquer le type de données JSON imbriqué, la colonne du point de terminaison source doit être renommée « json_ContactDetails », soit en ajoutant manuellement le préfixe « *json_* » sur la source, soit par le biais de règles de transformation. Par exemple, vous pouvez utiliser la règle de transformation ci-dessous :

```
{
    "rules": [
    {
    "rule-type": "transformation",
    "rule-id": "1",
    "rule-name": "1",
    "rule-target": "column",
    "object-locator": {
    "schema-name": "%",
    "table-name": "%",
    "column-name": "ContactDetails"
     },
    "rule-action": "rename",
    "value": "json_ContactDetails",
    "old-value": null
    }
    ]
```

```
}
```

AWS DMS réplique le champ ContactDetails au format JSON imbriqué, comme suit.

```
{
    "_id": "1",
    "FirstName": "John",
    "LastName": "Doe",
    "ContactDetails": {
        "Home": {
            "Address": "Boston",
            "Phone": "1111111111"
        },
        "Work": {
            "Address": "Boston",
            "Phone": "2222222222"
        }
    }
}
```

Utilisation d'un tableau JSON (préfixe array_)

Pour imposer un type de données, vous pouvez préfixer un nom de colonne avec `array_` (c'est-à-dire, `array_`*`columnName`*), manuellement ou à l'aide d'une transformation. Dans ce cas, AWS DMS considère la colonne comme un tableau JSON et le crée en tant que tel dans le document cible.

Supposons que vous souhaitiez migrer le document suivant à partir d'un point de terminaison source MongoDB.

```
{
    "_id" : "1",
    "FirstName": "John",
    "LastName": "Doe",

    "ContactAddresses": ["Boston", "New York"],

    "ContactPhoneNumbers": ["1111111111", "2222222222"]
}
```

Si vous n'imposez aucun des types de données source, le document `ContactDetails` imbriqué est migré sous la forme d'une chaîne.

```
{
    "_id": "1",
    "FirstName": "John",
    "LastName": "Doe",

    "ContactAddresses": "[\"Boston\", \"New York\"]",

    "ContactPhoneNumbers": "[\"1111111111\", \"2222222222\"]"

}
```

Toutefois, vous pouvez ajouter des règles de transformation pour forcer l'insertion de `ContactAddress` et de `ContactPhoneNumbers` dans des tableaux JSON, comme illustré dans le tableau suivant.

Nom de la colonne source d'origine	Colonne source renommée
`ContactAddress`	`array_ContactAddress`
`ContactPhoneNumbers`	`array_ContactPhoneNumbers`

AWS DMS réplique `ContactAddress` et `ContactPhoneNumbers` comme suit.

```
{
    "_id": "1",
    "FirstName": "John",
    "LastName": "Doe",
    "ContactAddresses": [
        "Boston",
        "New York"
    ],
    "ContactPhoneNumbers": [
        "1111111111",
        "2222222222"
    ]
}
```

Connexion à Amazon DocumentDB à l'aide de TLS

Par défaut, un cluster Amazon DocumentDB récemment créé n'accepte que les connexions sécurisées qui utilisent le protocole TLS (Transport Layer Security). Lorsque TLS est activé, chaque connexion à Amazon DocumentDB nécessite une clé publique.

Vous pouvez récupérer la clé publique pour Amazon DocumentDB en téléchargeant le fichier `rds-combined-ca-bundle.pem` à partir d'un compartiment Amazon S3 hébergé par AWS. Pour plus d'informations sur le téléchargement de ce fichier, consultez Chiffrement des connexions à l'aide de TLS dans le Guide du développeur Amazon DocumentDB.

Après avoir téléchargé ce fichier .pem, vous pouvez importer la clé publique qu'il contient dans AWS DMS, comme décrit ci-dessous.

AWS Management Console

Pour importer le fichier de clé publique (.pem)

1. Ouvrez la console AWS DMS à l'adresse https://console.aws.amazon.com/dms.
2. Dans le volet de navigation, choisissez Certificates.
3. Choisissez l'onglet Import certificate (Importer le certificat) et effectuez les opérations suivantes :
   - Pour Certificate identifier (Identifiant de certificat), entrez un nom unique pour le certificat, par exemple, `docdb-cert`.
   - Pour Import file (Importer un fichier), accédez à l'emplacement où vous avez enregistré le fichier .pem.

   Lorsque les paramètres vous conviennent, choisissez Add new CA certificate (Ajouter un nouveau certificat CA).

AWS CLI

Utilisez la commande `aws dms import-certificate`, comme illustré dans l'exemple suivant.

```
aws dms import-certificate \
    --certificate-identifier docdb-cert \
    --certificate-pem file://./rds-combined-ca-bundle.pem
```

Lorsque vous créez un point de terminaison cible AWS DMS, fournissez l'identifiant de certificat (par exemple, `docdb-cert`). Définissez aussi le paramètre de mode SSL sur `verify-full`.

Connexion aux clusters Elastic Amazon DocumentDB en tant que cible

Dans AWS DMS versions 3.4.7 et ultérieures, vous pouvez créer un point de terminaison cible Amazon DocumentDB en tant que cluster Elastic. Si vous créez votre point de terminaison cible en tant que cluster Elastic, vous devez attacher un nouveau certificat SSL au point de terminaison de votre cluster Elastic Amazon DocumentDB, car votre certificat SSL existant ne fonctionnera pas.

Pour attacher un nouveau certificat SSL au point de terminaison de votre cluster Elastic Amazon DocumentDB

1. Dans un navigateur, ouvrez https://www.amazontrust.com/repository/SFSRootCAG2.pem et enregistrez le contenu dans un fichier `.pem` portant un nom de fichier unique, par exemple `SFSRootCAG2.pem`. Il s'agit du fichier de certificat que vous devrez importer lors des étapes suivantes.
2. Créez le point de terminaison de cluster Elastic et définissez les options suivantes :
    a. Sous Configuration du point de terminaison, choisissez Ajouter un certificat CA.
    b. Pour Identifiant de certificat, entrez **`SFSRootCAG2.pem`**.
    c. Pour Importer un fichier de certificat, choisissez Choisir un fichier et accédez au fichier `SFSRootCAG2.pem` que vous avez téléchargé précédemment.
    d. Sélectionnez et ouvrez le fichier `SFSRootCAG2.pem` téléchargé.
    e. Sélectionnez Importer un certificat.
    f. Dans le menu déroulant Choisir un certificat, choisissez SFSRootCAG2.pem.

Le nouveau certificat SSL issu du fichier `SFSRootCAG2.pem` téléchargé est désormais attaché au point de terminaison de votre cluster Elastic Amazon DocumentDB.

Réplication continue avec Amazon DocumentDB en tant que cible

Si la réplication continue (capture des données de modification, CDC) est activée pour Amazon DocumentDB en tant que cible, AWS DMS versions 3.5.0 et ultérieures fournit une amélioration des performances vingt fois supérieure à celle des versions précédentes. Dans les versions précédentes où AWS DMS gérait jusqu'à 250 enregistrements par seconde, AWS DMS gère désormais environ 5 000 enregistrements/seconde. AWS DMS garantit également que les documents d'Amazon

DocumentDB restent synchronisés avec la source. Lorsqu'un enregistrement source est créé ou mis à jour, AWS DMS doit d'abord déterminer quel enregistrement Amazon DocumentDB est affecté en procédant comme suit :

- Si l'enregistrement source comporte une colonne nommée `_id`, la valeur de cette colonne détermine la valeur `_id` correspondante dans la collection Amazon DocumentDB.
- S'il n'y a pas de colonne `_id`, mais que les données source comportent une clé primaire ou un index unique, AWS DMS utilise cette valeur de clé ou d'index comme valeur `_id` pour la collection Amazon DocumentDB.
- Si l'enregistrement source ne comporte pas de colonne `_id`, ni de clé primaire ou d'index unique, AWS DMS fait correspondre toutes les colonnes source avec les champs correspondants dans la collection Amazon DocumentDB.

Lorsqu'un nouvel enregistrement source est créé, AWS DMS écrit un document correspondant dans Amazon DocumentDB. Si un enregistrement source existant est mis à jour, AWS DMS met à jour les champs correspondants du document cible dans Amazon DocumentDB. Tous les champs qui existent dans le document cible, mais pas dans l'enregistrement source, restent intacts.

Lorsqu'un enregistrement source est supprimé, AWS DMS supprime le document correspondant d'Amazon DocumentDB.

Modifications structurelles (DDL) à la source

Avec la réplication continue, les modifications des structures de données source (telles que les tables, colonnes, etc.) sont propagées à leurs homologues dans Amazon DocumentDB. Dans les bases de données relationnelles, ces modifications sont lancées à l'aide d'instructions DDL (Data Definition Language). Vous pouvez voir la façon dont AWS DMS propage ces modifications dans Amazon DocumentDB dans le tableau suivant.

DDL à la source	Effet sur la cible Amazon DocumentDB
`CREATE TABLE`	Crée une collection vide.
Déclaration qui renomme une table (`RENAME TABLE`, `ALTER TABLE...RENAME` , etc.)	Renomme la collection.
`TRUNCATE TABLE`	Supprime tous les documents de la collection, mais uniquement si `HandleSourceTableT`

DDL à la source	Effet sur la cible Amazon DocumentDB
	`runcated` a pour valeur `true`. Pour de plus amples informations, veuillez consulter Paramètres de tâche pour la gestion du DDL de traitement des modifications.
`DROP TABLE`	Supprime la collection, mais uniquement si `HandleSourceTableDropped` a pour valeur `true`. Pour de plus amples informations, veuillez consulter Paramètres de tâche pour la gestion du DDL de traitement des modifications.
Instruction qui ajoute une colonne à une table (`ALTER TABLE...ADD` , etc.)	L'instruction DDL est ignorée et un avertissement est émis. Lorsque la première instruction `INSERT` est exécutée à la source, le nouveau champ est ajouté au document cible.
`ALTER TABLE...RENAME COLUMN`	L'instruction DDL est ignorée et un avertissement est émis. Lorsque la première instruction `INSERT` est exécutée à la source, le nouveau champ est ajouté au document cible.
`ALTER TABLE...DROP COLUMN`	L'instruction DDL est ignorée et un avertissement est émis.
Déclaration qui modifie le type de données de la colonne (`ALTER COLUMN...MODIFY` , etc.)	L'instruction DDL est ignorée et un avertissement est émis. Lorsque la première instruction `INSERT` est exécutée à la source avec le nouveau type de données, le document cible est créé avec un champ ayant ce nouveau type de données.

Limitations de l'utilisation d'Amazon DocumentDB en tant que cible

Les limitations suivantes s'appliquent lorsque vous utilisez Amazon DocumentDB en tant que cible pour AWS DMS :

- Dans Amazon DocumentDB, les noms de collection ne peuvent pas contenir le symbole du dollar ($). En outre, les noms de base de données ne peuvent pas contenir de caractères Unicode.
- AWS DMS ne prend pas en charge la fusion de plusieurs tables source dans une seule collection Amazon DocumentDB.
- Lorsque AWS DMS traite les modifications provenant d'une table source qui ne possède pas de clé primaire, les colonnes LOB de cette table sont ignorées.
- Si l'option Change table (Modifier la table) est activée et que AWS DMS rencontre une colonne source nommée « _id », cette colonne apparaît sous la forme « __id » (deux traits de soulignement) dans la table modifiée.
- Si vous choisissez Oracle comme point de terminaison source, une journalisation supplémentaire complète doit être activée sur la source Oracle. Dans le cas contraire, s'il existe des colonnes à la source qui n'ont pas été modifiées, les données seront chargées dans Amazon DocumentDB tant que valeurs nulles.
- Le paramètre de tâche de réplication `TargetTablePrepMode:TRUNCATE_BEFORE_LOAD` n'est pas pris en charge pour une utilisation avec un point de terminaison cible DocumentDB.

Utilisation des paramètres de point de terminaison avec Amazon DocumentDB en tant que cible

Vous pouvez utiliser des paramètres de point de terminaison pour configurer la base de données cible Amazon DocumentDB comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison cible à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--doc-db-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec Amazon DocumentDB en tant que cible sont indiqués dans le tableau suivant.

Nom d'attribut	Valeurs valides	Valeur par défaut et description

`replicateShardCollections`	boolean
	
	`true`
	
	`false`
		Lorsque ce paramètre de point de terminaison est défini sur `true`, il a les effets suivants et impose les limitations suivantes :
		
		• AWS DMS est autorisé à répliquer les données vers des collections de partitions cibles. Ce paramètre

Nom d'attribut	Valeurs valides	Valeur par défaut et description
		n'est applicable que si le point de terminaison DocumentDB cible est un cluster Elastic. • Vous devez définir `TargetTablePrepMode` sur `DO_NOTHING` . • AWS DMS définit automatiquement `useUpdate Lookup` sur `false` pendant la migration.

Types de données cibles pour Amazon DocumentDB

Dans le tableau suivant, vous trouverez les types de données cibles Amazon DocumentDB qui sont pris en charge lorsque vous utilisez AWS DMS et le mappage par défaut à partir des types de données AWS DMS. Pour plus d'informations sur les types de données AWS DMS, consultez Types de données pour AWS Database Migration Service.

Type de données AWS DMS	Type de données Amazon DocumentDB
BOOLEAN	Booléen
BYTES	Données binaires
DATE	Date
TIME	Chaîne (UTF8)
DATETIME	Date
INT1	Entier 32 bits
INT2	Entier 32 bits
INT4	Entier 32 bits
INT8	Entier 64 bits
NUMERIC	Chaîne (UTF8)

Type de données AWS DMS	Type de données Amazon DocumentDB
REAL4	Double
REAL8	Double
CHAÎNE	Si les données sont reconnues comme étant au format JSON, AWS DMS les migre vers Amazon DocumentDB sous la forme d'un document. Dans le cas contraire, les données sont mappées au format Chaîne (UTF8).
UINT1	Entier 32 bits
UINT2	Entier 32 bits
UINT4	Entier 64 bits
UINT8	Chaîne (UTF8)
WSTRING	Si les données sont reconnues comme étant au format JSON, AWS DMS les migre vers Amazon DocumentDB sous la forme d'un document. Dans le cas contraire, les données sont mappées au format Chaîne (UTF8).
BLOB	Binaire
CLOB	Si les données sont reconnues comme étant au format JSON, AWS DMS les migre vers Amazon DocumentDB sous la forme d'un document. Dans le cas contraire, les données sont mappées au format Chaîne (UTF8).
NCLOB	Si les données sont reconnues comme étant au format JSON, AWS DMS les migre vers Amazon DocumentDB sous la forme d'un document. Dans le cas contraire, les données sont mappées au format Chaîne (UTF8).

Utilisation d'Amazon Neptune en tant que cible pour AWS Database Migration Service

Amazon Neptune est un service de base de données orientée graphe entièrement géré et fiable, qui facilite la création et l'exécution d'applications fonctionnant avec des jeux de données hautement connectés. Le cœur de Neptune est un moteur de base de données orientée graphe spécialisé et hautes performances. Ce moteur est optimisé pour le stockage de milliards de relations et l'interrogation du graphe avec une latence de l'ordre de quelques millisecondes. Neptune prend en charge les langages de requête de graphe populaires Apache TinkerPop Gremlin et SPARQL de W3C. Pour plus d'informations sur Amazon Neptune, consultez Qu'est-ce qu'Amazon Neptune ? dans le Guide de l'utilisateur Amazon Neptune.

Sans une base de données orientée graphe telle que Neptune, vous modélisez probablement des données hautement connectées dans une base de données relationnelle. Étant donné que les données ont des connexions potentiellement dynamiques, les applications qui utilisent de telles sources de données doivent modéliser les requêtes de données connectées dans SQL. Cette approche nécessite que vous écriviez une couche supplémentaire pour convertir les requêtes de graphe en SQL. En outre, les bases de données relationnelles sont dotées d'une rigidité de schéma. Toute modification du schéma pour modéliser les connexions changeantes nécessite des temps d'arrêt et une maintenance supplémentaire de la conversion de requête pour prendre en charge le nouveau schéma. Les performances des requêtes sont également une autre contrainte importante à prendre en compte lors de la conception de vos applications.

Les bases de données graphiques peuvent grandement simplifier de telles situations. Libre d'un schéma, une couche de requête graphique enrichie (Gremlin ou SPARQL) et des index optimisés pour les requêtes graphiques augmentent la flexibilité et les performances. La base de données orientée graphe Amazon Neptune comporte également des fonctionnalités d'entreprise telles que le chiffrement au repos, une couche d'autorisation sécurisée, les sauvegardes par défaut, la prise en charge de Multi-AZ, la prise en charge de réplicas en lecture, etc.

À l'aide d'AWS DMS, vous pouvez migrer des données relationnelles qui modélisent un graphe hautement connecté vers un point de terminaison cible Neptune à partir d'un point de terminaison source DMS pour toute base de données SQL prise en charge.

Pour plus de détails, consultez les sections suivantes.

Rubriques

- Vue d'ensemble de la migration vers Amazon Neptune en tant que cible

- Spécification des paramètres de point de terminaison pour Amazon Neptune en tant que cible
- Création d'une fonction de service IAM pour l'accès à Amazon Neptune en tant que cible
- Spécification de règles de mappage de graphe à l'aide de Gremlin et de R2RML pour Amazon Neptune en tant que cible
- Types de données pour la migration Gremlin et R2RML vers Amazon Neptune en tant que cible
- Limitations de l'utilisation d'Amazon Neptune en tant que cible

Vue d'ensemble de la migration vers Amazon Neptune en tant que cible

Avant de commencer une migration vers une cible Neptune, créez les ressources suivantes dans votre compte AWS :

- Un cluster Neptune pour le point de terminaison cible.
- Une base de données relationnelle SQL prise en charge par AWS DMS pour le point de terminaison source.
- Un compartiment Amazon S3 pour le point de terminaison cible. Créez ce compartiment S3 dans la même région AWS que votre cluster Neptune. AWS DMS utilise ce compartiment S3 comme stockage de fichiers intermédiaires pour les données cibles qu'il charge en bloc dans la base de données Neptune. Pour plus d'informations sur la création d'un compartiment S3, consultez Créer un compartiment dans le Guide de l'utilisateur Amazon Simple Storage Service.
- Un point de terminaison de cloud privé virtuel (VPC) pour S3 dans le même VPC que le cluster Neptune.
- Rôle AWS Identity and Access Management (IAM) qui inclut une stratégie IAM. Cette stratégie doit spécifier les autorisations `GetObject`, `PutObject`, `DeleteObject` et `ListObject` pour le compartiment S3 pour votre point de terminaison cible. Ce rôle est assumé par AWS DMS et Neptune avec un accès IAM à la fois au compartiment S3 cible et à la base de données Neptune. Pour de plus amples informations, veuillez consulter Création d'une fonction de service IAM pour l'accès à Amazon Neptune en tant que cible.

Une fois ces ressources disponibles, la configuration et le démarrage d'une migration vers une cible Neptune sont similaires à toute migration de chargement complet à l'aide de la console ou de l'API DMS. Toutefois, une migration vers une cible Neptune nécessite des étapes uniques.

Pour migrer une base de données relationnelle AWS DMS vers Neptune

1. Créez une instance de réplication comme décrit dans Création d'une instance de réplication.
2. Créez et testez une base de données relationnelle SQL prise en charge par AWS DMS pour le point de terminaison source.
3. Créez et testez le point de terminaison cible pour la base de données Neptune.

   Pour connecter le point de terminaison cible à la base de données Neptune, spécifiez le nom du serveur pour le point de terminaison du cluster Neptune ou le point de terminaison de l'instance d'écriture Neptune. Spécifiez également le dossier du compartiment S3 pour AWS DMS afin de stocker ses fichiers intermédiaires en vue d'un chargement en bloc dans la base de données Neptune.

   Pendant la migration, AWS DMS stocke toutes les données cibles migrées dans ce dossier compartiment S3 jusqu'à la taille maximale de fichier que vous spécifiez. Lorsque ce stockage de fichiers atteint cette taille maximale, AWS DMS charge en bloc les données S3 stockées dans la base de données cible. Il efface le dossier pour permettre le stockage de toutes les données cibles supplémentaires pour le chargement ultérieur dans la base de données cible. Pour de plus amples informations sur la spécification de ces paramètres, veuillez consulter Spécification des paramètres de point de terminaison pour Amazon Neptune en tant que cible.
4. Créez une tâche de réplication à chargement complet avec les ressources créées aux étapes 1 à 3 et procédez comme suit :

   a. Utilisez le mappage de tables de tâches comme d'habitude pour identifier des schémas, tables et vues sources spécifiques à migrer depuis votre base de données relationnelle à l'aide de règles de sélection et de transformation appropriées. Pour de plus amples informations, veuillez consulter Utilisation du mappage de table pour spécifier des paramètres de tâche.

   b. Spécifiez les mappages de cible en choisissant l'une des options suivantes pour spécifier les règles de mappage à partir de tables et de vues sources vers votre graphe de base de données cible Neptune :

      - Gremlin JSON : pour en savoir plus sur l'utilisation de Gremlin JSON pour charger une base de données Neptune, consultez Format des données de chargement Gremlin dans le Guide de l'utilisateur Amazon Neptune.

- SPARQL RDB to Resource Description Framework Mapping Language (R2RML) : pour en savoir plus sur l'utilisation de SPARQL R2RML, consultez la spécification W3C R2RML : langage de mappage RDB à RDF (langue française non garantie).

c. Effectuez l'une des actions suivantes :

- À l'aide de la console AWS DMS, spécifiez les options de mappage de graphique en utilisant les règles de mappage de graphique sur la page Créer une tâche de migration de base de données.
- À l'aide de l'API AWS DMS, spécifiez ces options à l'aide du paramètre de demande `TaskData` de l'appel d'API `CreateReplicationTask`.

Pour de plus amples informations et consulter des exemples utilisant Gremlin JSON et SPARQL R2RML afin de spécifier des règles de mappage graphique, veuillez consulter Spécification de règles de mappage de graphe à l'aide de Gremlin et de R2RML pour Amazon Neptune en tant que cible.

5. Démarrez la réplication de votre tâche de migration.

Spécification des paramètres de point de terminaison pour Amazon Neptune en tant que cible

Pour créer ou modifier un point de terminaison cible, vous pouvez utiliser la console, `CreateEndpoint` ou les opérations d'API `ModifyEndpoint`.

Pour une cible Neptune dans la console AWS DMS, spécifiez les Paramètres spécifiques au point de terminaison sur la page Créer un point de terminaison ou Modifier un point de terminaison de la console. Pour `CreateEndpoint` et `ModifyEndpoint`, spécifiez les paramètres de demande pour l'option `NeptuneSettings`. L'exemple suivant montre comment procéder à l'aide de l'interface de ligne de commande.

```
dms create-endpoint --endpoint-identifier my-neptune-target-endpoint
--endpoint-type target --engine-name neptune
--server-name my-neptune-db.cluster-cspckvklbvgf.us-east-1.neptune.amazonaws.com
--port 8192
--neptune-settings
     '{"ServiceAccessRoleArn":"arn:aws:iam::123456789012:role/myNeptuneRole",
       "S3BucketName":"my-bucket",
       "S3BucketFolder":"my-bucket-folder",
```

```
        "ErrorRetryDuration":57,
        "MaxFileSize":100,
        "MaxRetryCount": 10,
        "IAMAuthEnabled":false}'
```

Ici, l'option `--server-name` de l'interface CLI spécifie le nom du serveur pour le point de terminaison d'écriture de cluster Neptune. Vous pouvez également spécifier le nom du serveur pour un point de terminaison d'instance d'écriture Neptune.

Les paramètres de demande d'option `--neptune-settings` sont les suivants :

- `ServiceAccessRoleArn` : (Obligatoire) Amazon Resource Name (ARN) de la fonction de service que vous avez créée pour le point de terminaison cible Neptune. Pour de plus amples informations, veuillez consulter Création d'une fonction de service IAM pour l'accès à Amazon Neptune en tant que cible.
- `S3BucketName` : (Obligatoire) nom du compartiment S3 dans lequel DMS peut stocker temporairement des données de graphe migrées dans des fichiers .csv avant de les charger en bloc dans la base de données cible Neptune. DMS mappe les données source SQL aux données de graphique avant de les stocker dans ces fichiers .csv.
- `S3BucketFolder` : (Obligatoire) chemin d'accès au dossier dans lequel vous souhaitez que DMS stocke les données de graphe migrées dans le compartiment S3 spécifié par `S3BucketName`.
- `ErrorRetryDuration` : (Facultatif) nombre de millisecondes pendant lesquelles DMS doit attendre pour réessayer une charge groupée de données de graphe migrées vers la base de données cible Neptune avant de déclencher une erreur. La valeur par défaut est 250.
- `MaxFileSize` : (Facultatif) taille maximale en Ko de données de graphe migrées stockées dans un fichier .csv avant que DMS charge en bloc les données dans la base de données cible Neptune. La valeur par défaut est de 1 048 576 Ko (1 Go). En cas de succès, DMS efface le compartiment, prêt à stocker le prochain lot de données de graphique migrées.
- `MaxRetryCount` : (Facultatif) nombre de fois où DMS peut réessayer une charge en bloc de données de graphe migrées vers la base de données cible Neptune avant de déclencher une erreur. La valeur par défaut est 5.
- `IAMAuthEnabled` : (Facultatif) si vous souhaitez que l'autorisation IAM soit activée pour ce point de terminaison, définissez ce paramètre sur `true` et attachez le document de politique IAM approprié à votre fonction de service spécifié par `ServiceAccessRoleArn`. La valeur par défaut est `false`.

Création d'une fonction de service IAM pour l'accès à Amazon Neptune en tant que cible

Pour accéder à Neptune en tant que cible, créez un rôle de service à l'aide d'IAM. Selon la configuration du point de terminaison Neptune, attachez à ce rôle une partie ou la totalité des documents de politique IAM et d'approbation suivants. Lorsque vous créez le point de terminaison Neptune, vous fournissez l'ARN de cette fonction de service. Cela permet à AWS DMS et Amazon Neptune d'assumer les autorisations d'accès à Neptune et à son compartiment Amazon S3 associé.

Si vous définissez le paramètre `IAMAuthEnabled` dans `NeptuneSettings` sur `true` dans votre configuration de point de terminaison Neptune, attachez une politique IAM telle que la suivante à votre fonction de service. Si vous définissez `IAMAuthEnabled` sur `false`, vous pouvez ignorer cette stratégie.

```
// Policy to access Neptune

    {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "VisualEditor0",
                "Effect": "Allow",
                "Action": "neptune-db:*",
                "Resource": "arn:aws:neptune-db:us-east-1:123456789012:cluster-
CLG7H7FHK54AZGHEH6MNS55JKM/*"
            }
        ]
    }
```

La politique IAM précédente permet un accès complet au cluster cible Neptune spécifié par `Resource`.

Joignez une stratégie IAM telle que la suivante à votre rôle de service. Cette stratégie permet à DMS de stocker temporairement des données de graphe migrées dans le compartiment S3 que vous avez créé pour le chargement en bloc dans la base de données cible Neptune.

```
//Policy to access S3 bucket

{
 "Version": "2012-10-17",
 "Statement": [{
```

```
    "Sid": "ListObjectsInBucket0",
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": [
     "arn:aws:s3:::my-bucket"
    ]
   },
   {
    "Sid": "AllObjectActions",
    "Effect": "Allow",
    "Action": ["s3:GetObject",
     "s3:PutObject",
     "s3:DeleteObject"
    ],

    "Resource": [
     "arn:aws:s3:::my-bucket/"
    ]
   },
   {
    "Sid": "ListObjectsInBucket1",
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": [
     "arn:aws:s3:::my-bucket",
     "arn:aws:s3:::my-bucket/"
    ]
   }
  ]
}
```

La politique IAM précédente permet à votre compte d'interroger le contenu du compartiment S3 (`arn:aws:s3:::my-bucket`) créé pour votre cible Neptune. Elle permet également à votre compte de fonctionner pleinement sur le contenu de tous les fichiers et dossiers de compartiment (`arn:aws:s3:::my-bucket/`).

Modifiez la relation d'approbation et attachez le rôle IAM suivant à votre fonction de service pour permettre à AWS DMS et au service de base de données Amazon Neptune d'assumer le rôle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "dms.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Sid": "neptune",
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Pour en savoir plus sur la spécification de cette fonction de service pour votre point de terminaison cible Neptune, consultez Spécification des paramètres de point de terminaison pour Amazon Neptune en tant que cible.

Spécification de règles de mappage de graphe à l'aide de Gremlin et de R2RML pour Amazon Neptune en tant que cible

Les règles de mappage de graphe que vous créez spécifient comment les données extraites d'une source de base de données relationnelle SQL sont chargées dans une cible de cluster de base de données Neptune. Le format de ces règles de mappage diffère selon que les règles servent à charger des données de graphique de propriétés à l'aide d'Apache TinkerPop Gremlin ou de données RDF (Resource Description Framework) à l'aide de R2RML. Vous trouverez ci-dessous des informations sur ces formats et sur les autres ressources disponibles.

Vous pouvez spécifier ces règles de mappage lorsque vous créez la tâche de migration à l'aide de la console ou de l'API DMS.

À l'aide de la console, spécifiez ces règles de mappage à l'aide des règles de mappage de graphique sur la page Créer une tâche de migration de base de données. Dans Règles de mappage de graphique, vous pouvez entrer et modifier les règles de mappage directement à l'aide de l'éditeur fourni. Vous pouvez également rechercher un fichier contenant les règles de mappage dans le format de mappage de graphique approprié.

À l'aide de l'API, spécifiez ces options à l'aide du paramètre de demande `TaskData` de l'appel d'API `CreateReplicationTask`. Définissez `TaskData` sur le chemin d'accès d'un fichier contenant les règles de mappage dans le format de mappage de graphique approprié.

Règles de mappage de graphique pour générer des données de graphique de propriétés à l'aide de Gremlin

À l'aide de Gremlin pour générer les données du graphique de propriétés, spécifiez un objet JSON avec une règle de mappage pour chaque entité de graphique à générer à partir des données source. Le format de ce code JSON est défini spécifiquement pour charger Amazon Neptune en bloc. Le modèle suivant montre à quoi ressemble chaque règle de cet objet.

```
{
    "rules": [
        {
            "rule_id": "(an identifier for this rule)",
            "rule_name": "(a name for this rule)",
            "table_name": "(the name of the table or view being loaded)",
            "vertex_definitions": [
                {
                    "vertex_id_template": "{col1}",
                    "vertex_label": "(the vertex to create)",
                    "vertex_definition_id": "(an identifier for this vertex)",
                    "vertex_properties": [
                        {
                            "property_name": "(name of the property)",
                            "property_value_template": "{col2} or text",
                            "property_value_type": "(data type of the property)"
                        }
                    ]
                }
            ]
        },
        {
            "rule_id": "(an identifier for this rule)",
            "rule_name": "(a name for this rule)",
            "table_name": "(the name of the table or view being loaded)",
            "edge_definitions": [
                {
                    "from_vertex": {
                        "vertex_id_template": "{col1}",
```

```
                        "vertex_definition_id": "(an identifier for the vertex
 referenced above)"
                    },
                    "to_vertex": {
                        "vertex_id_template": "{col3}",
                        "vertex_definition_id": "(an identifier for the vertex
 referenced above)"
                    },
                    "edge_id_template": {
                        "label": "(the edge label to add)",
                        "template": "{col1}_{col3}"
                    },
                    "edge_properties":[
                        {
                            "property_name": "(the property to add)",
                            "property_value_template": "{col4} or text",
                            "property_value_type": "(data type like String, int,
 double)"
                        }
                    ]
                }
            ]
        }
    ]
}
```

La présence d'une étiquette de sommet implique que le sommet est créé ici. Son absence implique que le sommet est créé par une source différente, et cette définition n'ajoute que des propriétés de sommet. Spécifiez autant de définitions de sommets et de tronçons que nécessaire pour spécifier les mappages pour l'ensemble de la source de base de données relationnelle.

Voici un exemple de règle pour une table `employee`.

```
{
    "rules": [
        {
            "rule_id": "1",
            "rule_name": "vertex_mapping_rule_from_nodes",
            "table_name": "nodes",
            "vertex_definitions": [
                {
```

```
                    "vertex_id_template": "{emp_id}",
                    "vertex_label": "employee",
                    "vertex_definition_id": "1",
                    "vertex_properties": [
                        {
                            "property_name": "name",
                            "property_value_template": "{emp_name}",
                            "property_value_type": "String"
                        }
                    ]
                }
            ]
        },
        {
            "rule_id": "2",
            "rule_name": "edge_mapping_rule_from_emp",
            "table_name": "nodes",
            "edge_definitions": [
                {
                    "from_vertex": {
                        "vertex_id_template": "{emp_id}",
                        "vertex_definition_id": "1"
                    },
                    "to_vertex": {
                        "vertex_id_template": "{mgr_id}",
                        "vertex_definition_id": "1"
                    },
                    "edge_id_template": {
                        "label": "reportsTo",
                        "template": "{emp_id}_{mgr_id}"
                    },
                    "edge_properties":[
                        {
                            "property_name": "team",
                            "property_value_template": "{team}",
                            "property_value_type": "String"
                        }
                    ]
                }
            ]
        }
    ]
}
```

Ici, les définitions de sommets et de tronçons mappent une relation de reporting à partir d'un nœud `employee` avec l'ID d'employé (`EmpID`) et d'un nœud `employee` avec un ID de responsable (`managerId`).

Pour plus d'informations sur la création de règles de mappage de graphe à l'aide de Gremlin JSON, consultez Format des données de chargement Gremlin dans le Guide de l'utilisateur Amazon Neptune.

Règles de mappage de graphique pour la génération de données RDF/SPARQL

Si vous chargez des données RDF à interroger à l'aide de SPARQL, écrivez les règles de mappage de graphique dans R2RML. R2RML est un langage W3C standard pour mapper des données relationnelles vers RDF. Dans un fichier R2RML, un mappage de triples (par exemple, `<#TriplesMap1>` suivant) spécifie une règle pour convertir chaque ligne d'une table logique en zéro ou plusieurs triples RDF. Un mappage de sujet (par exemple, tout élément `rr:subjectMap` suivant) spécifie une règle pour générer les sujets des triples RDF générés par un mappage de triples. Un mappage prédicat-objet (par exemple, tout élément `rr:predicateObjectMap` suivant) est une fonction qui crée une ou plusieurs paires prédicat-objet pour chaque ligne de table logique d'une table logique.

Voici un exemple simple de table `nodes`.

```
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix ex: <http://example.com/ns#>.

<#TriplesMap1>
    rr:logicalTable [ rr:tableName "nodes" ];
    rr:subjectMap [
        rr:template "http://data.example.com/employee/{id}";
        rr:class ex:Employee;
    ];
    rr:predicateObjectMap [
        rr:predicate ex:name;
        rr:objectMap [ rr:column "label" ];
    ]
```

Dans l'exemple précédent, le mappage définit des nœuds de graphique mappés à partir d'une table d'employés.

Voici un autre exemple simple de table `Student`.

```
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix ex: <http://example.com/#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

<#TriplesMap2>
    rr:logicalTable [ rr:tableName "Student" ];
    rr:subjectMap   [ rr:template "http://example.com/{ID}{Name}";
                      rr:class foaf:Person ];
    rr:predicateObjectMap [
        rr:predicate ex:id ;
        rr:objectMap  [ rr:column "ID";
                        rr:datatype xsd:integer ]
    ];
    rr:predicateObjectMap [
        rr:predicate foaf:name ;
        rr:objectMap  [ rr:column "Name" ]
    ].
```

Dans l'exemple précédent, le mappage définit des nœuds de graphique qui mappent des relations de type ami d'un ami entre des personnes d'une table `Student`.

Pour de plus amples informations sur la création de règles de mappage de graphique à l'aide de SPARQL R2RML, veuillez consulter la spécification W3C R2RML : RDB to RDF Mapping Language.

Types de données pour la migration Gremlin et R2RML vers Amazon Neptune en tant que cible

AWS DMS effectue le mappage de type de données du point de terminaison source SQL vers votre cible Neptune de l'une des deux manières suivantes. La solution choisie dépend du format de mappage de graphe que vous utilisez pour charger la base de données Neptune :

- Apache TinkerPop Gremlin, en utilisant une représentation JSON des données de migration.
- SPARQL du W3C, en utilisant une représentation R2RML des données de migration.

Pour de plus amples informations sur ces deux formats de mappage de graphique, veuillez consulter Spécification de règles de mappage de graphe à l'aide de Gremlin et de R2RML pour Amazon Neptune en tant que cible.

Vous trouverez ci-dessous des descriptions des mappages de types de données pour chaque format.

Mappages de type de données cible source SQL vers Gremlin

Le tableau suivant présente les mappages de type de données d'une source SQL vers une cible formatée Gremlin.

AWS DMS mappe tout type de données source SQL non répertorié à un Gremlin `String`.

Types de données sources SQL	Types de données cibles Gremlin
`NUMERIC` (et variantes)	`Double`
`DECIMAL`	
`TINYINT`	`Byte`
`SMALLINT`	`Short`
`INT, INTEGER`	`Int`
`BIGINT`	`Long`
`FLOAT`	`Float`
`DOUBLE PRECISION`	
`REAL`	`Double`
`BIT`	`Boolean`
`BOOLEAN`	
`DATE`	`Date`
`TIME`	
`TIMESTAMP`	
`CHARACTER` (et variantes)	`String`

Pour plus d'informations sur les types de données Gremlin pour le chargement de Neptune, consultez Types de données Gremlin dans le Guide de l'utilisateur Neptune.

Mappages de type de données cible Source SQL vers R2RML (RDF)

Le tableau suivant présente les mappages de types de données d'une source SQL vers une cible formatée R2RML.

Tous les types de données RDF répertoriés sont sensibles à la casse, à l'exception du littéral RDF. AWS DMS mappe tout type de données source SQL non répertorié à un littéral RDF.

Un littéral RDF est l'une des diverses formes lexicales littérales et l'un des différents types de données. Pour de plus amples informations, veuillez consulter Littéraux RDF dans la spécification W3C Resource Description Framework (RDF) : Concepts and Abstract Syntax.

Types de données sources SQL	Types de données cibles R2RML (RDF)
BINARY (et variantes)	xsd:hexBinary
NUMERIC (et variantes)	xsd:decimal
DECIMAL	
TINYINT	xsd:integer
SMALLINT	
INT, INTEGER	
BIGINT	
FLOAT	xsd:double
DOUBLE PRECISION	
REAL	
BIT	xsd:boolean
BOOLEAN	
DATE	xsd:date

Types de données sources SQL	Types de données cibles R2RML (RDF)
`TIME`	`xsd:time`
`TIMESTAMP`	`xsd:dateTime`
`CHARACTER` (et variantes)	Littéral RDF

Pour plus d'informations sur les types de données RDF pour charger Neptune et leurs mappages aux types de données sources SQL, consultez Conversions de type de données dans la spécification W3C R2RML : langage de mappage RDB à RDF (langue française non garantie).

Limitations de l'utilisation d'Amazon Neptune en tant que cible

Les limitations suivantes s'appliquent lorsque vous utilisez Neptune en tant que cible :

- AWS DMS prend actuellement en charge les tâches de chargement complet uniquement pour la migration vers une cible Neptune. La migration CDC (capture des données de modification) vers une cible Neptune n'est pas prise en charge.
- Assurez-vous d'effacer manuellement toutes les données de la base de données Neptune cible avant de commencer la tâche de migration, comme dans les exemples suivants.

  Pour supprimer toutes les données (sommets et arêtes) dans le graphe, exécutez la commande Gremlin suivante.

  ```
  gremlin> g.V().drop().iterate()
  ```

  Pour supprimer les sommets portant l'étiquette `'customer'`, exécutez la commande Gremlin suivante.

  ```
  gremlin> g.V().hasLabel('customer').drop()
  ```

  > **Note**
  >
  > La suppression d'un jeu de données volumineux peut prendre un certain temps. Vous pouvez vouloir itérer `drop()` avec une limite, par exemple `limit(1000)`.

Pour supprimer les arêtes portant l'étiquette `'rated'`, exécutez la commande Gremlin suivante.

```
gremlin> g.E().hasLabel('rated').drop()
```

> **Note**
>
> La suppression d'un jeu de données volumineux peut prendre un certain temps. Vous pouvez vouloir itérer `drop()` avec une limite, par exemple `limit(1000)`.

- L'opération d'API DMS `DescribeTableStatistics` peut renvoyer des résultats inexacts sur une table donnée en raison de la nature des structures de données de graphe Neptune.

  Pendant la migration, AWS DMS analyse chaque table source et utilise le mappage de graphe pour convertir les données source en graphe Neptune. Les données converties sont d'abord stockées dans le dossier de compartiment S3 spécifié pour le point de terminaison cible. Si la source est analysée et que ces données S3 intermédiaires sont générées avec succès, `DescribeTableStatistics` suppose que les données ont été chargées avec succès dans la base de données cible Neptune. Mais ce n'est pas toujours vrai. Pour vérifier que les données ont été chargées correctement pour une table donnée, comparez les valeurs `count()` renvoyées aux deux extrémités de la migration pour cette table.

  Dans l'exemple suivant, AWS DMS a chargé une table `customer` à partir de la base de données source, à laquelle est affectée l'étiquette `'customer'` dans le graphe de base de données Neptune cible. Vous pouvez vous assurer que cette étiquette est écrite dans la base de données cible. Pour ce faire, comparez le nombre de lignes `customer` disponibles à partir de la base de données source avec le nombre de lignes étiquetées `'customer'` chargées dans la base de données cible Neptune une fois la tâche terminée.

  Pour obtenir le nombre de lignes client disponibles à partir de la base de données source à l'aide de SQL, exécutez la procédure suivante.

  ```
  select count(*) from customer;
  ```

  Pour obtenir le nombre de lignes étiquetées `'customer'` chargées dans le graphique de base de données cible à l'aide de Gremlin, exécutez la procédure suivante.

```
gremlin> g.V().hasLabel('customer').count()
```

- Actuellement, si une seule table n'est pas chargée, c'est toute la tâche qui échoue. Contrairement à une cible de base de données relationnelle, les données dans Neptune sont hautement connectées, ce qui rend impossible dans de nombreux cas la reprise d'une tâche. Si une tâche ne peut pas être reprise correctement en raison de ce type d'échec de chargement des données, créez une nouvelle tâche pour charger la table qui n'a pas pu être chargée. Avant d'exécuter cette nouvelle tâche, effacez manuellement la table partiellement chargée de la cible Neptune.

  > **Note**
  >
  > Vous pouvez reprendre une tâche dont la migration vers une cible Neptune a échoué si l'échec est récupérable (par exemple, une erreur de transit réseau).

- AWS DMS prend en charge la plupart des normes pour R2RML. Cependant, AWS DMS ne prend pas en charge certaines normes R2RML, y compris les expressions inverses, les jointures et les vues. Une solution de contournement pour une vue R2RML consiste à créer une vue SQL personnalisée correspondante dans la base de données source. Dans la tâche de migration, utilisez le mappage de table pour choisir la vue en tant qu'entrée. Mappez ensuite la vue à une table qui est ensuite consommée par R2RML pour générer des données de graphique.
- Lorsque vous migrez des données sources avec des types de données SQL non pris en charge, les données cibles résultantes peuvent avoir une perte de précision. Pour de plus amples informations, veuillez consulter Types de données pour la migration Gremlin et R2RML vers Amazon Neptune en tant que cible.
- AWS DMS ne prend pas en charge la migration des données LOB dans une cible Neptune.

Utilisation de Redis en tant que cible pour AWS Database Migration Service

Redis est un magasin de structures de données open source en mémoire, utilisé comme base de données, cache et agent de messages. La gestion des données en mémoire peut entraîner des opérations de lecture ou d'écriture de moins d'une milliseconde et des centaines de millions d'opérations par seconde. En tant que magasin de données en mémoire, Redis alimente les applications les plus exigeantes nécessitant des temps de réponse inférieurs à la milliseconde.

Avec AWS DMS, vous pouvez migrer des données à partir de n'importe quelle base de données source prise en charge vers un magasin de données Redis cible, avec un temps d'arrêt minimal. Pour plus d'informations sur Redis, consultez la documentation de Redis.

Outre Redis sur site, AWS Database Migration Service prend en charge les systèmes suivants :

- Amazon ElastiCache for Redis en tant que magasin de données cible. ElastiCache for Redis fonctionne avec vos clients Redis et utilise le format de données ouvert pour stocker vos données.
- Amazon MemoryDB for Redis en tant que magasin de données cible. MemoryDB est compatible avec Redis et vous permet de créer des applications en utilisant toutes les structures de données, API et commandes Redis utilisées aujourd'hui.

Pour plus d'informations sur l'utilisation de Redis en tant que cible pour AWS DMS, consultez les sections suivantes :

Rubriques

- Conditions préalables à l'utilisation d'un cluster Redis en tant que cible pour AWS DMS
- Limitations de l'utilisation de Redis en tant que cible pour AWS Database Migration Service
- Migration de données d'une base de données relationnelle ou non relationnelle vers une cible Redis
- Spécification des paramètres de point de terminaison pour Redis en tant que cible

Conditions préalables à l'utilisation d'un cluster Redis en tant que cible pour AWS DMS

DMS prend en charge une cible Redis sur site dans une configuration autonome, ou en tant que cluster Redis dans lequel les données sont automatiquement partitionnées sur plusieurs nœuds. Le partitionnement est le processus qui consiste à séparer les données en petits fragments appelés « partitions », répartis sur plusieurs serveurs ou nœuds. Une partition contient un sous-ensemble de l'ensemble de données total qui couvre une partie de la charge de travail globale.

Étant donné que Redis est un magasin de données NoSQL de type clé-valeur, la convention de dénomination des clés Redis à utiliser lorsque votre source est une base de données relationnelle est nom-schéma.nom-table.clé-primaire. Dans Redis, la clé et la valeur ne doivent pas contenir le caractère spécial %. Dans le cas contraire, DMS ignore l'enregistrement.

> **Note**
>
> Si vous utilisez ElastiCache for Redis en tant que cible, DMS prend uniquement en charge les configurations compatibles avec le mode cluster. Pour plus d'informations sur l'utilisation d'ElastiCache for Redis version 6.x ou versions ultérieures afin de créer un magasin de données cible compatible avec le mode cluster, consultez Mise en route dans le Guide de l'utilisateur Amazon ElastiCache for Redis.

Avant de commencer une migration de base de données, lancez votre cluster Redis selon les critères suivants.

- Votre cluster possède une ou plusieurs partitions.
- Si vous utilisez une cible ElastiCache for Redis, assurez-vous que votre cluster n'utilise pas le contrôle d'accès basé sur les rôles IAM. Utilisez plutôt Redis Auth pour authentifier les utilisateurs.
- Activez Multi-AZ (zones de disponibilité).
- Assurez-vous que le cluster dispose de suffisamment de mémoire pour accueillir les données à migrer depuis la base de données.
- Assurez-vous que votre cluster Redis cible ne comporte plus aucune donnée avant de commencer la tâche de migration initiale.

Vous devez déterminer vos exigences de sécurité pour la migration des données avant de créer la configuration de votre cluster. DMS prend en charge la migration vers les groupes de réplication cibles, quelle que soit leur configuration de chiffrement. Mais vous ne pouvez activer ou désactiver le chiffrement que lorsque vous créez la configuration de votre cluster.

Limitations de l'utilisation de Redis en tant que cible pour AWS Database Migration Service

Les limitations suivantes s'appliquent lorsque vous utilisez Redis en tant que cible :

- Étant donné que Redis est un magasin de données no-sql de type clé-valeur, la convention de dénomination des clés Redis à utiliser lorsque votre source est une base de données relationnelle est `schema-name.table-name.primary-key`.
- Dans Redis, la paire clé-valeur ne peut pas contenir le caractère spécial `%`. Dans le cas contraire, DMS ignore l'enregistrement.

- DMS ne migrera pas les lignes contenant des caractères spéciaux.
- DMS ne migrera pas les champs dont le nom contient des caractères spéciaux.
- Le Mode LOB complet n'est pas pris en charge.
- Une autorité de certification (CA) privée n'est pas prise en charge lors de l'utilisation d'ElastiCache for Redis en tant que cible.

Migration de données d'une base de données relationnelle ou non relationnelle vers une cible Redis

Vous pouvez migrer les données de n'importe quel magasin de données SQL ou NoSQL source directement vers une cible Redis. La configuration et le démarrage d'une migration vers une cible Redis sont similaires à n'importe quelle migration de chargement complet et de CDC à l'aide de la console ou de l'API DMS. Pour effectuer une migration de base de données vers une cible Redis, procédez comme suit.

- Créez une instance de réplication pour effectuer tous les processus de la migration. Pour de plus amples informations, veuillez consulter Création d'une instance de réplication.
- Spécifiez un point de terminaison source. Pour de plus amples informations, veuillez consulter Création de points de terminaison source et cible.
- Trouvez le numéro de port et le nom DNS de votre cluster.
- Téléchargez un ensemble de certificats que vous pouvez utiliser pour vérifier les connexions SSL.
- Spécifiez un point de terminaison cible, comme décrit ci-dessous.
- Créez une tâche ou un ensemble de tâches pour définir les tables et processus de réplication à utiliser. Pour de plus amples informations, veuillez consulter Création d'une tâche.
- Migrez les données de la base de données source vers votre cluster cible.

Commencez une migration de base de données de l'une des deux manières suivantes :

1. Vous pouvez choisir la console AWS DMS et y effectuer chaque étape.
2. Vous pouvez utiliser AWS Command Line Interface (AWS CLI). Pour plus d'informations sur l'utilisation de l'interface de ligne de commande avec AWS DMS, consultez AWS CLI pour AWS DMS.

Pour trouver le numéro de port et le nom DNS de votre cluster

- Utilisez la commande AWS CLI suivante pour définir `replication-group-id` sur le nom de votre groupe de réplication.

```
aws elasticache describe-replication-groups --replication-group-id myreplgroup
```

Ici, la sortie indique le nom DNS dans l'attribut `Address` et le numéro de port dans l'attribut `Port` du nœud primaire du cluster.

```
 ...
"ReadEndpoint": {
"Port": 6379,
"Address": "myreplgroup-
111.1abc1d.1111.uuu1.cache.example.com"
}
...
```

Si vous utilisez MemoryDB for Redis en tant que cible, utilisez la commande AWS CLI suivante pour fournir une adresse de point de terminaison à votre cluster Redis.

```
aws memorydb describe-clusters --clusterid clusterid
```

Téléchargez un ensemble de certificats à utiliser pour vérifier les connexions SSL.

- Entrez la commande `wget` suivante sur la ligne de commande. Wget est un utilitaire de ligne de commande GNU gratuit qui permet de télécharger des fichiers à partir d'Internet.

```
wget https://s3.aws-api-domain/rds-downloads/rds-combined-ca-bundle.pem
```

Ici, *`aws-api-domain`* indique le domaine Amazon S3 de votre région AWS requis pour accéder au compartiment S3 spécifié et au fichier rds-combined-ca-bundle.pem qu'il fournit.

Pour créer un point de terminaison cible à l'aide de la console AWS DMS

Ce point de terminaison est destiné à votre cible Redis déjà en cours d'exécution.

- Dans la console, choisissez Points de terminaison dans le volet de navigation, puis Créer un point de terminaison. Le tableau suivant décrit les paramètres.

Pour cette option	Faire ceci
Type de point de terminaison	Choisissez le type de point de terminaison Cible.
Identificateur de point de terminaison	Entrez le nom du point de terminaison. Indiquez le type de point de terminaison dans le nom, par exemple **`my-redis-target`** .
Moteur cible	Choisissez Redis comme type de moteur de base de données auquel vous souhaitez que ce point de terminaison se connecte.
Nom du cluster	Entrez le nom DNS de votre cluster Redis.
Port	Entrez le numéro de port de votre cluster Redis.
Protocole de sécurité SSL	Choisissez Texte brut ou Chiffrement SSL. Texte brut : cette option ne fournit pas le chiffrement du protocole TLS (Transport Layer Security) pour le trafic entre le point de terminaison et la base de données. Chiffrement SSL : si vous choisissez cette option, entrez un ARN de certificat de CA SSL pour vérifier le certificat du serveur et établir une connexion chiffrée. Pour Redis sur site, DMS prend en charge les autorités de certification (CA) publiques et privées. Pour ElastiCache for Redis, DMS ne prend en charge qu'une CA publique.

Pour cette option	Faire ceci
Type d'authentification	Choisissez le type d'authentification à effectuer lors de la connexion à Redis. Les options incluent Aucun, Rôle d'authentification et Jeton d'authentification. Si vous choisissez Rôle d'authentification, entrez un Nom d'utilisateur d'authentification et un Mot de passe d'authentification. Si vous choisissez Jeton d'authentification, fournissez uniquement un Mot de passe d'authentification.
Instance de réplication	[Facultatif] Uniquement si vous prévoyez de tester votre connexion, choisissez le nom de l'instance de réplication que vous avez entré précédemment sur la page Créer une instance de réplication.

Lorsque vous avez terminé de fournir toutes les informations relatives au point de terminaison, AWS DMS crée votre point de terminaison cible Redis à utiliser lors de la migration de la base de données.

Pour en savoir plus sur la création d'une tâche de migration et le démarrage de la migration de la base de données, consultez Création d'une tâche.

Spécification des paramètres de point de terminaison pour Redis en tant que cible

Pour créer ou modifier un point de terminaison cible, vous pouvez utiliser la console, `CreateEndpoint` ou les opérations d'API `ModifyEndpoint`.

Pour une cible Redis dans la console AWS DMS, spécifiez les Paramètres spécifiques du point de terminaison sur la page Créer un point de terminaison ou Modifier un point de terminaison de la console.

Lorsque vous utilisez les opérations d'API `CreateEndpoint` et `ModifyEndpoint`, spécifiez les paramètres de demande pour l'option `RedisSettings`. L'exemple suivant montre comment procéder à l'aide d'AWS CLI.

```
aws dms create-endpoint --endpoint-identifier my-redis-target
--endpoint-type target --engine-name redis --redis-settings
```

```
'{"ServerName":"sample-test-
sample.zz012zz.cluster.eee1.cache.bbbxxx.com","Port":6379,"AuthType":"auth-token",
 "SslSecurityProtocol":"ssl-encryption", "AuthPassword":"notanactualpassword"}'

{
    "Endpoint": {
        "EndpointIdentifier": "my-redis-target",
        "EndpointType": "TARGET",
        "EngineName": "redis",
        "EngineDisplayName": "Redis",
        "TransferFiles": false,
        "ReceiveTransferredFiles": false,
        "Status": "active",
        "KmsKeyId": "arn:aws:kms:us-east-1:999999999999:key/x-b188188x",
        "EndpointArn": "arn:aws:dms:us-
east-1:555555555555:endpoint:ABCDEFGHIJKLMONOPQRSTUVWXYZ",
        "SslMode": "none",
        "RedisSettings": {
            "ServerName": "sample-test-sample.zz012zz.cluster.eee1.cache.bbbxxx.com",
            "Port": 6379,
            "SslSecurityProtocol": "ssl-encryption",
            "AuthType": "auth-token"
        }
    }
}
```

Les paramètres `--redis-settings` sont les suivants :

- `ServerName` : (Obligatoire) pour le type `string`, spécifie le cluster Redis vers lequel les données seront migrées et qui se trouve dans votre VPC.
- `Port` : (Obligatoire) pour le type `number`, valeur du port utilisé pour accéder au point de terminaison.
- `SslSecurityProtocol` : (Facultatif) les valeurs valides incluent `plaintext` et `ssl-encryption`. La valeur par défaut est `ssl-encryption`.

  L'option `plaintext` ne fournit pas le chiffrement du protocole TLS (Transport Layer Security) pour le trafic entre le point de terminaison et la base de données.

  Utilisez `ssl-encryption` pour établir une connexion chiffrée. `ssl-encryption` ne requiert pas d'ARN d'autorité de certification (CA) SSL pour vérifier le certificat d'un serveur, mais il est possible

d'en identifier un à l'aide du paramètre `SslCaCertificateArn`. Si aucun ARN de CA n'est fourni, DMS utilise la CA racine Amazon.

Lorsque vous utilisez une cible Redis sur site, vous pouvez utiliser `SslCaCertificateArn` pour importer une autorité de certification (CA) publique ou privée dans DMS et fournir cet ARN pour l'authentification du serveur. Une CA privée n'est pas prise en charge lors de l'utilisation d'ElastiCache for Redis en tant que cible.

- `AuthType` : (Obligatoire) indique le type d'authentification à effectuer lors de la connexion à Redis. Les valeurs valides sont `none`, `auth-token` et `auth-role`.

  L'option `auth-token` nécessite qu'un « *`AuthPassword`* » soit fourni, tandis que l'option `auth-role` nécessite que « *`AuthUserName`* » et « *`AuthPassword`* » soient fournis.

Utilisation de Babelfish en tant que cible pour AWS Database Migration Service

Vous pouvez migrer des données d'une base de données source Microsoft SQL Server vers une cible Babelfish à l'aide d'AWS Database Migration Service.

Babelfish pour Aurora PostgreSQL étend l'Édition compatible avec PostgreSQL de votre base de données Amazon Aurora et permet d'accepter les connexions de base de données à partir de clients Microsoft SQL Server. Les applications initialement conçues pour SQL Server peuvent ainsi être utilisées avec Aurora PostgreSQL, sans apporter de changements significatifs au code par rapport à une migration traditionnelle et sans modifier les pilotes de base de données.

Pour en savoir plus sur les versions de Babelfish qui sont prises en charge par AWS DMS en tant que cible, consultez Objectifs pour AWS DMS. Les versions antérieures de Babelfish sur Aurora PostgreSQL nécessitent une mise à niveau avant d'utiliser le point de terminaison Babelfish.

> **Note**
>
> Le point de terminaison cible Aurora PostgreSQL est la méthode privilégiée pour migrer des données vers Babelfish. Pour de plus amples informations, veuillez consulter Utilisation de Babelfish for Aurora PostgreSQL en tant que cible.

Pour en savoir plus sur l'utilisation de Babelfish comme point de terminaison de base de données, consultez Babelfish for Aurora PostgreSQL dans le Guide de l'utilisateur Amazon Aurora pour Aurora.

Prérequis pour l'utilisation de Babelfish en tant que cible pour AWS DMS

Vous devez créer vos tables avant de migrer les données afin de vous assurer qu'AWS DMS utilise les types de données et les métadonnées de table appropriés. Si vous ne créez pas vos tables sur la cible avant d'exécuter la migration, AWS DMS risque de créer les tables avec des autorisations et des types de données incorrects. Par exemple, AWS DMS crée une colonne d'horodatage au format binary(8) et ne fournit pas la fonctionnalité timestamp/rowversion attendue.

Pour préparer et créer vos tables avant la migration

1. Exécutez vos instructions DDL de création de table qui incluent des contraintes uniques, des clés primaires ou des contraintes par défaut.

   N'incluez pas de contraintes de clé étrangère ni d'instructions DDL pour des objets tels que des vues, des procédures stockées, des fonctions ou des déclencheurs. Vous pouvez les appliquer après avoir migré la base de données source.
2. Identifiez les colonnes d'identité, les colonnes calculées ou les colonnes contenant des types de données rowversion ou timestamp pour vos tables. Créez ensuite les règles de transformation nécessaires pour gérer les problèmes connus lors de l'exécution de la tâche de migration. Pour plus d'informations, consultez Règles et actions de transformation.
3. Identifiez les colonnes contenant des types de données que Babelfish ne prend pas en charge. Modifiez ensuite les colonnes concernées dans la table cible pour utiliser les types de données pris en charge, ou créez une règle de transformation qui les supprime lors de la tâche de migration. Pour plus d'informations, consultez Règles et actions de transformation.

   Le tableau suivant répertorie les types de données source non pris en charge par Babelfish, ainsi que le type de données cible recommandé correspondant à utiliser.

Type de données source	Type de données Babelfish recommandé
HEIRARCHYID	NVARCHAR(250)
GEOMETRY	VARCHAR(MAX)
GEOGRAPHY	VARCHAR(MAX)

Pour définir le niveau des unités de capacité Aurora (ACU) pour la base de données source Aurora PostgreSQL sans serveur V2

Vous pouvez améliorer les performances de votre tâche de migration AWS DMS avant de l'exécuter en définissant la valeur minimale d'ACU.

- Dans la fenêtre Paramètres de capacité Severless v2, définissez Nombre minimal d'ACU sur **2** ou sur un niveau raisonnable pour votre cluster de bases de données Aurora.

  Pour plus d'informations sur la définition des unités de capacité Aurora, consultez Choix de la plage de capacité Aurora sans serveur v2 pour un cluster Aurora dans le Guide de l'utilisateur Amazon Aurora.

Après avoir exécuté votre tâche de migration AWS DMS, vous pouvez réinitialiser la valeur minimale de vos ACU à un niveau raisonnable pour la base de données source Aurora PostgreSQL sans serveur V2.

Exigences de sécurité lors de l'utilisation de Babelfish en tant que cible pour AWS Database Migration Service

La section suivante décrit les exigences de sécurité pour l'utilisation d'AWS DMS avec une cible Babelfish :

- Le nom d'utilisateur de l'administrateur (utilisateur Admin) utilisé pour créer la base de données.
- L'identifiant de connexion PSQL et l'utilisateur disposant des autorisations SELECT, INSERT, UPDATE, DELETE et REFERENCES suffisantes.

Autorisations utilisateur pour utiliser Babelfish en tant que cible pour AWS DMS

> **Important**
>
> À des fins de sécurité, le compte d'utilisateur utilisé pour la migration des données doit être un utilisateur enregistré dans une base de données Babelfish que vous utilisez en tant que cible.

Votre point de terminaison cible Babelfish nécessite des autorisations utilisateur minimales pour exécuter une migration AWS DMS.

Pour créer un identifiant de connexion et un utilisateur Transact-SQL (T-SQL) à faible privilège

1. Créez un identifiant de connexion et un mot de passe à utiliser lors de la connexion au serveur.

```
CREATE LOGIN dms_user WITH PASSWORD = 'password';
GO
```

2. Créez la base de données virtuelle pour votre cluster Babelfish.

```
CREATE DATABASE my_database;
GO
```

3. Créez l'utilisateur T-SQL pour la base de données cible.

```
USE my_database
GO
CREATE USER dms_user FOR LOGIN dms_user;
GO
```

4. Pour chaque table de la base de données Babelfish, accordez (GRANT) des autorisations aux tables.

```
GRANT SELECT, DELETE, INSERT, REFERENCES, UPDATE ON [dbo].[Categories] TO dms_user;
```

Limitations de l'utilisation de Babelfish en tant que cible pour AWS Database Migration Service

Les limitations suivantes s'appliquent lors de l'utilisation d'une base de données Babelfish en tant que cible pour AWS DMS :

- Seul le mode de préparation de table « Ne rien faire » est pris en charge.
- Le type de données ROWVERSION nécessite une règle de mappage de table qui supprime le nom de colonne de la table pendant la tâche de migration.
- Le type de données sql_variant n'est pas pris en charge.
- Le mode LOB complet est pris en charge. L'utilisation de SQL Server comme point de terminaison source nécessite que l'attribut de connexion au point de terminaison SQL Server `ForceFullLob=True` soit défini afin que les objets LOB soient migrés vers le point de terminaison cible.

- Les paramètres de la tâche de réplication présentent les limitations suivantes :

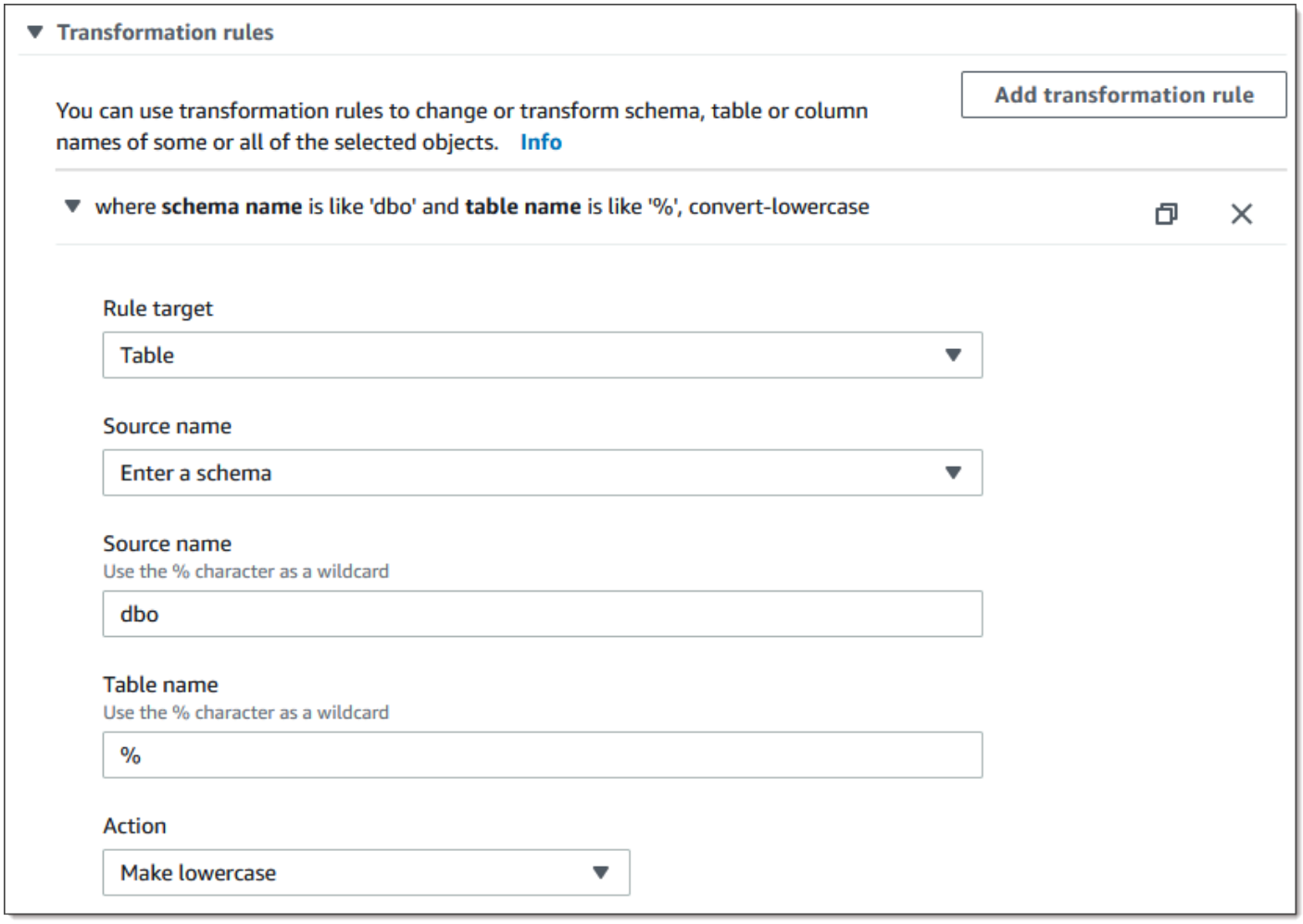
```
{
   "FullLoadSettings": {
      "TargetTablePrepMode": "DO_NOTHING",
      "CreatePkAfterFullLoad": false,
      }.

}
```

- Les types de données TIME(7), DATETIME2(7) et DATETIMEOFFSET(7) dans Babelfish limitent la valeur de précision pour les secondes à 6 chiffres. Envisagez d'utiliser une valeur de précision de 6 pour votre table cible lorsque vous utilisez ces types de données. Pour Babelfish versions 2.2.0 et ultérieures, lorsque vous utilisez TIME(7) et DATETIME2(7), le septième chiffre de précision est toujours zéro.
- En mode DO_NOTHING, DMS vérifie si la table existe déjà. Si la table n'existe pas dans le schéma cible, DMS crée la table en fonction de la définition de table source et mappe tous les types de données définis par l'utilisateur à leur type de données de base.
- Une tâche de migration AWS DMS vers une cible Babelfish ne prend pas en charge les tables dont les colonnes utilisent les types de données ROWVERSION ou TIMESTAMP. Vous pouvez utiliser une règle de mappage de table qui supprime le nom de colonne de la table pendant le processus de transfert. Dans l'exemple de règle de transformation suivant, une table nommée `Actor` dans votre source est transformée pour supprimer toutes les colonnes commençant par les caractères `col` de la table `Actor` dans votre cible.

```
{
  "rules": [{
  "rule-type": "selection",is
  "rule-id": "1",
  "rule-name": "1",
  "object-locator": {
   "schema-name": "test",
   "table-name": "%"
  },
  "rule-action": "include"
 }, {
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
```

```
  "rule-action": "remove-column",
  "rule-target": "column",
  "object-locator": {
   "schema-name": "test",
   "table-name": "Actor",
   "column-name": "col%"
  }
 }]
 }
```

- Pour les tables comportant des colonnes d'identité ou calculées, où les tables cibles utilisent des noms en casse mixte tels que Catégories, vous devez créer une action de règle de transformation qui convertit les noms des table en minuscules pour votre tâche DMS. L'exemple suivant illustre la création de l'action de règle de transformation Mettre en minuscules à l'aide de la console AWS DMS. Pour de plus amples informations, veuillez consulter Règles et actions de transformation.



- Avant Babelfish version 2.2.0, DMS limitait le nombre de colonnes que vous pouviez répliquer vers un point de terminaison cible Babelfish à vingt (20) colonnes. Avec Babelfish 2.2.0, la limite est

passée à 100 colonnes. Mais avec Babelfish versions 2.4.0 et ultérieures, le nombre de colonnes que vous pouvez répliquer augmente à nouveau. Vous pouvez exécuter l'exemple de code suivant sur la base de données SQL Server pour déterminer quelles tables sont trop longues.

```
USE myDB;
GO
DECLARE @Babelfish_version_string_limit INT = 8000; -- Use 380 for Babelfish versions
 before 2.2.0
WITH bfendpoint
AS (
SELECT
 [TABLE_SCHEMA]
      ,[TABLE_NAME]
   , COUNT( [COLUMN_NAME] ) AS NumberColumns
   , ( SUM( LEN( [COLUMN_NAME] ) + 3)
  + SUM( LEN( FORMAT(ORDINAL_POSITION, 'N0') ) + 3 )
     + LEN( TABLE_SCHEMA ) + 3
  + 12 -- INSERT INTO string
  + 12)  AS InsertIntoCommandLength -- values string
      , CASE WHEN ( SUM( LEN( [COLUMN_NAME] ) + 3)
  + SUM( LEN( FORMAT(ORDINAL_POSITION, 'N0') ) + 3 )
     + LEN( TABLE_SCHEMA ) + 3
  + 12 -- INSERT INTO string
  + 12)  -- values string
   >= @Babelfish_version_string_limit
   THEN 1
   ELSE 0
  END AS IsTooLong
FROM [INFORMATION_SCHEMA].[COLUMNS]
GROUP BY [TABLE_SCHEMA], [TABLE_NAME]
)
SELECT *
FROM bfendpoint
WHERE IsTooLong = 1
ORDER BY TABLE_SCHEMA, InsertIntoCommandLength DESC, TABLE_NAME
;
```

Types de données cibles pour Babelfish

Le tableau suivant présente les types de données cibles Babelfish qui sont pris en charge lorsque vous utilisez AWS DMS et le mappage par défaut à partir des types de données AWS DMS.

Pour de plus amples informations sur les types de données AWS DMS, veuillez consulter Types de données pour AWS Database Migration Service.

Type de données AWS DMS	Type de données Babelfish
BOOLEAN	TINYINT
BYTES	VARBINARY(Length)
DATE	DATE
TIME	TIME
INT1	SMALLINT
INT2	SMALLINT
INT4	INT
INT8	BIGINT
NUMERIC	NUMERIC(p,s)
REAL4	REAL
REAL8	FLOAT
CHAÎNE	Si la colonne est une colonne de date ou d'heure, effectuez les opérations suivantes : • Pour SQL Server 2008 et versions ultérieures, utilisez DATETIME2. • Pour les versions antérieures, si l'échelle est 3 ou inférieure utilisez DATETIME. Dans tous les autres cas, utilisez VARCHAR (37). Si la colonne n'est pas une colonne de date ou d'heure, utilisez VARCHAR (length).
UINT1	TINYINT

Type de données AWS DMS	Type de données Babelfish
UINT2	SMALLINT
UINT4	INT
UINT8	BIGINT
WSTRING	NVARCHAR(length)
BLOB	VARBINARY(max) Pour pouvoir utiliser ce type de données avec DMS, vous devez activer l'utilisation des types de données BLOB pour une tâche spécifique. DMS prend en charge les types de données BLOB uniquement dans les tables qui contiennent une clé primaire.
CLOB	VARCHAR(max) Pour pouvoir utiliser ce type de données avec DMS, vous devez activer l'utilisation des types de données CLOB pour une tâche spécifique.
NCLOB	NVARCHAR(max) Pour pouvoir utiliser ce type de données avec DMS, vous devez activer l'utilisation des types de données NCLOB pour une tâche spécifique. Au cours de la CDC, DMS prend en charge les types de données NCLOB uniquement dans les tables qui contiennent une clé primaire.

Utilisation d'Amazon Timestream comme cible pour AWS Database Migration Service

Vous pouvez utiliser AWS Database Migration Service pour migrer les données de votre base de données source vers un point de terminaison cible Amazon Timestream, en prenant en charge les migrations à chargement complet et CDC.

Amazon Timestream est un service de base de données de séries temporelles rapide, évolutif et sans serveur, conçu pour l'ingestion de grands volumes de données. Les données de séries temporelles sont une séquence de points de données collectés pendant un intervalle de temps et utilisés pour mesurer des événements qui évoluent au fil du temps. Il est utilisé pour collecter, stocker et analyser les métriques des applications IoT, des DevOps applications et des applications d'analyse. Une fois que vous avez enregistré vos données dans Timestream, vous pouvez visualiser et identifier des tendances et des modèles dans vos données en temps quasi réel. Pour plus d'informations sur Amazon Timestream, consultez Qu'est-ce qu'Amazon Timestream ? dans le Guide du développeur Amazon Timestream.

Rubriques

- Conditions préalables à l'utilisation d'Amazon Timestream comme cible pour AWS Database Migration Service
- Paramètres de tâche de chargement complet multithread
- Paramètres de tâche de chargement CDC multithread
- Paramètres de point de terminaison lors de l'utilisation de Timestream comme cible pour AWS DMS
- Création et modification d'un point de terminaison Amazon Timestream cible
- Utilisation du mappage d'objet pour migrer des données vers une rubrique Timestream
- Limites d'utilisation d'Amazon Timestream comme cible pour AWS Database Migration Service

Conditions préalables à l'utilisation d'Amazon Timestream comme cible pour AWS Database Migration Service

Avant de configurer Amazon Timestream comme cible pour AWS DMS, veillez à créer un rôle IAM. Ce rôle doit permettre à AWS DMS d'accéder aux données en cours de migration vers Amazon Timestream. L'ensemble d'autorisations d'accès minimum pour le rôle que vous utilisez pour migrer vers Timestream figure dans la politique IAM suivante.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDescribeEndpoints",
            "Effect": "Allow",
            "Action": [
                "timestream:DescribeEndpoints"
            ],
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "timestream:ListTables",
                "timestream:DescribeDatabase"
            ],
            "Resource": "arn:aws:timestream:region:account_id:database/DATABASE_NAME"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "timestream:DeleteTable",
                "timestream:WriteRecords",
                "timestream:UpdateTable",
                "timestream:CreateTable"
            ],
            "Resource": "arn:aws:timestream:region:account_id:database/DATABASE_NAME/
table/TABLE_NAME"
        }
    ]
}
```

Si vous avez l'intention de migrer toutes les tables, utilisez * pour *TABLE_NAME* dans l'exemple ci-dessus.

Notez ce qui suit à propos de l'utilisation de Timestream comme cible :

- Si vous avez l'intention d'ingérer des données historiques dont l'horodatage remonte à plus d'un an, nous vous recommandons d'utiliser AWS DMS pour écrire les données dans Amazon S3

au format CSV (valeurs séparées par des virgules). Utilisez ensuite le chargement par lots de Timestream pour ingérer les données dans Timestream. Pour plus d'informations, consultez Utilisation du chargement par lots dans Timestream dans le Guide du développeur Amazon Timestream.

- Pour les migrations de données à chargement complet datant de moins d'un an, nous recommandons de définir une période de conservation de la mémoire de la table Timestream supérieure ou égale à l'horodatage le plus ancien. Ensuite, une fois la migration terminée, modifiez la période de conservation de la mémoire de la table à la valeur souhaitée. Par exemple, pour migrer des données dont l'horodatage le plus ancien date de 2 mois, procédez comme suit :
  - Définissez la période de conservation de la mémoire de la table cible Timestream à 2 mois.
  - Démarrez la migration des données à l'aide d'AWS DMS.
  - Une fois la migration des données terminée, modifiez la période de conservation de la table Timestream cible à la valeur souhaitée.

  Nous vous recommandons d'estimer le coût de stockage de la mémoire avant la migration en utilisant les informations des pages suivantes :
  - Tarification d'Amazon Timestream
  - AWS Pricing Calculator
- Pour les migrations de données CDC, nous recommandons de définir la période de rétention de la mémoire de la table cible de manière à ce que les données ingérées se situent dans les limites de cette période. Pour plus d'informations, consultez Bonnes pratiques en matière d'écriture dans le Guide du développeur Amazon Timestream.

Paramètres de tâche de chargement complet multithread

Pour accélérer le transfert de données, AWS DMS prend en charge une tâche de chargement complet multithread vers un point de terminaison cible Timestream avec les paramètres de tâche suivants :

- `MaxFullLoadSubTasks` : utilisez cette option pour indiquer le nombre maximal de tables sources à charger en parallèle. DMS charge chaque table dans sa table Amazon Timestream cible correspondante à l'aide d'une sous-tâche dédiée. La valeur par défaut est 8 ; la valeur maximale 49.
- `ParallelLoadThreads` : utilisez cette option pour spécifier le nombre de threads utilisés par AWS DMS pour charger chaque table dans sa table cible Amazon Timestream. La valeur maximale

pour une cible Timestream est 32. Vous pouvez demander une augmentation de cette limite maximale.

- `ParallelLoadBufferSize` : utilisez cette option pour spécifier le nombre maximal d'enregistrements à stocker dans la mémoire tampon utilisée par les threads de chargement parallèles pour charger les données dans la cible Amazon Timestream. La valeur par défaut est 50. La valeur maximale est 1 000. Utilisez ce paramètre avec `ParallelLoadThreads`. `ParallelLoadBufferSize` est valide uniquement dans le cas de plusieurs threads.
- `ParallelLoadQueuesPerThread` : utilisez cette option pour spécifier le nombre de files d'attente auxquelles chaque thread simultané accède pour extraire les enregistrements de données des files d'attente et générer un chargement par lots pour la cible. La valeur par défaut est 1. Toutefois, pour les cibles Amazon Timestream de différentes tailles de charge utile, la plage valide est comprise entre 5 et 512 files d'attente par thread.

Paramètres de tâche de chargement CDC multithread

Pour accroître les performances de la fonctionnalité CDC, AWS DMS prend en charge les paramètres de tâche suivants :

- `ParallelApplyThreads` : spécifie le nombre de threads simultanés utilisés par AWS DMS pendant un chargement CDC pour transférer des enregistrements de données vers un point de terminaison cible Timestream. La valeur par défaut est 0 et la valeur maximale est 32.
- `ParallelApplyBufferSize` : spécifie le nombre maximal d'enregistrements à stocker dans chaque file d'attente de la mémoire tampon pour que les threads simultanés soient transférés vers un point de terminaison cible Timestream lors d'un chargement CDC. La valeur par défaut est 100 et la valeur maximale est 1 000. Utilisez cette option lorsque `ParallelApplyThreads` spécifie plusieurs threads.
- `ParallelApplyQueuesPerThread` : spécifie le nombre de files d'attente auxquelles chaque thread accède pour extraire les enregistrements de données et générer un chargement par lots pour un point de terminaison Timestream lors d'un chargement CDC. La valeur par défaut est 1 et la valeur maximale est 512.

Paramètres de point de terminaison lors de l'utilisation de Timestream comme cible pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer votre base de données Timestream cible comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez

les paramètres lorsque vous créez le point de terminaison cible à l'aide de la console AWS DMS ou à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--timestream-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec Timestream comme cible sont indiqués dans le tableau suivant.

Name (Nom)	Description
`MemoryDuration`	Définissez cet attribut pour spécifier la période de conservation des données migrées dans la mémoire de Timestream. Le temps est exprimé en heures. La mémoire de Timestream est optimisée pour un débit d'ingestion élevé et un accès rapide. Valeur par défaut : 24 (heures) Valeurs valides : 1 à 8 736 (1 heure à 12 mois exprimés en heures) Exemple : `--timestream-settings '{"MemoryDuration": 20}'`
`DatabaseName`	Définissez cet attribut pour spécifier le nom de la base de données Timestream cible. Type : chaîne Exemple : `--timestream-settings '{"DatabaseName": " db_name"}'`
`TableName`	Définissez cet attribut pour spécifier le nom de la table Timestream cible. Type : chaîne Exemple : `--timestream-settings '{"TableName": " table_name "}'`

Name (Nom)	Description
`MagneticDuration`	Définissez cet attribut pour spécifier la période de conservation du stockage magnétique appliquée aux tables Timestream (en jours). Il s'agit de la durée limite de conservation des données ingérées. Timestream supprime tout horodatage dépassant la durée limite de conservation. Pour plus d'informations, consultez Stockage dans le Guide du développeur Amazon Timestream. Exemple : `--timestream-settings '{"MagneticDuration": "3"}'`
`CdcInsertsAndUpdates`	Définissez cet attribut sur `true` pour spécifier qu'AWS DMS applique uniquement des insertions et des mises à jour, et pas de suppressions. Timestream n'autorise pas la suppression des enregistrements. Par conséquent, si cette valeur est `false`, AWS DMS annule l'enregistrement correspondant dans la base de données Timestream au lieu de le supprimer. Pour plus d'informations, consultez Limites, ci-après. Valeur par défaut : `false` Exemple : `--timestream-settings '{"CdcInsertsAndUpdates": "true"}'`

Name (Nom)	Description
`EnableMagneticStoreWrites`	Définissez cet attribut sur `true` pour activer les écritures sur stockage magnétique. Lorsque cette valeur est définie sur `false`, AWS DMS n'écrit pas les enregistr ements dont l'horodatage est antérieur à la période de conservation de la mémoire de la table cible, car Timestream n'autorise pas les écritures sur stockage magnétique par défaut. Pour plus d'informations, consultez Bonnes pratiques en matière d'écriture dans le Guide du développeur Amazon Timestream. Valeur par défaut : `false` Exemple : `--timestream-settings '{"Enable MagneticStoreWrites": "true"}'`

Création et modification d'un point de terminaison Amazon Timestream cible

Une fois que vous avez créé un rôle IAM et défini l'ensemble d'autorisations d'accès minimum, vous pouvez créer un point de terminaison Amazon Timestream cible à l'aide de la console AWS DMS ou de la commande `create-endpoint` d'AWS CLI, avec la syntaxe JSON `--timestream-settings '{"EndpointSetting": "value", ...}'`.

Les exemples suivants montrent comment créer et modifier un point de terminaison Timestream cible à l'aide d'AWS CLI.

Commande de création d'un point de terminaison Timestream cible

```
aws dms create-endpoint –endpoint-identifier timestream-target-demo
--endpoint-type target –engine-name timestream
--service-access-role-arn arn:aws:iam::123456789012:role/my-role
--timestream-settings
{
    "MemoryDuration": 20,
    "DatabaseName":"db_name",
    "MagneticDuration": 3,
    "CdcInsertsAndUpdates": true,
    "EnableMagneticStoreWrites": true,
```

```
}
```

Commande de modification d'un point de terminaison Timestream cible

```
aws dms modify-endpoint –endpoint-identifier timestream-target-demo
--endpoint-type target –engine-name timestream
--service-access-role-arn arn:aws:iam::123456789012:role/my-role
--timestream-settings
{
    "MemoryDuration": 20,
    "MagneticDuration": 3,
}
```

Utilisation du mappage d'objet pour migrer des données vers une rubrique Timestream

AWS DMS utilise les règles de mappage de table pour mapper les données de la source à la rubrique Timestream cible. Pour mapper des données à une rubrique cible, vous utilisez un type de règle de mappage de table appelé « mappage d'objet ». Vous utilisez le mappage d'objet pour définir la façon dont les enregistrements de données de la source sont mappés aux enregistrements de données publiés dans une rubrique Timestream.

Les rubriques Timestream ne disposent pas d'une structure prédéfinie autre que le fait d'avoir une clé de partition.

> **Note**
>
> Vous n'avez pas besoin d'utiliser le mappage d'objet. Vous pouvez utiliser un mappage de table standard pour différentes transformations. Cependant, le type de clé de partition suivra les comportements par défaut suivants :
>
> - La clé primaire est utilisée comme clé de partition pour le chargement complet.
> - Si aucun paramètre de tâche parallel-apply n'est utilisé, `schema.table` est utilisé comme clé de partition pour la CDC.
> - Si des paramètres de tâche parallel-apply sont utilisés, la clé primaire est utilisée comme clé de partition pour la CDC.

Pour créer une règle de mappage d'objet, spécifiez `rule-type` comme `object-mapping`. Cette règle spécifie le type de mappage d'objet que vous souhaitez utiliser. La structure de la règle est la suivante.

```
{
    "rules": [
        {
            "rule-type": "object-mapping",
            "rule-id": "id",
            "rule-name": "name",
            "rule-action": "valid object-mapping rule action",
            "object-locator": {
                "schema-name": "case-sensitive schema name",
                "table-name": ""
            }
        }
    ]
}
```

```
{
    "rules": [
        {
            "rule-type": "object-mapping",
            "rule-id": "1",
            "rule-name": "timestream-map",
            "rule-action": "map-record-to-record",
            "target-table-name": "tablename",
            "object-locator": {
                "schema-name": "",
                "table-name": ""
            },
            "mapping-parameters": {
                "timestream-dimensions": [
                    "column_name1",
                     "column_name2"
                ],
                "timestream-timestamp-name": "time_column_name",
                "timestream-multi-measure-name": "column_name1or2",
                "timestream-hash-measure-name":  true or false,
                "timestream-memory-duration": x,
                "timestream-magnetic-duration": y
            }
```

```
        }
    ]
}
```

AWS DMS prend actuellement en charge `map-record-to-record` et `map-record-to-document` comme les seules valeurs valides pour le paramètre `rule-action`. Les valeurs `map-record-to-record` et `map-record-to-document` spécifient l'action par défaut d'AWS DMS sur les enregistrements qui ne sont pas exclus en raison de leur appartenance à la liste des attributs `exclude-columns`. Ces valeurs n'affectent en aucune façon les mappages d'attributs.

Utilisez `map-record-to-record` lorsque vous migrez une base de données relationnelle vers une rubrique Timestream. Ce type de règle utilise la valeur `taskResourceId.schemaName.tableName` de la base de données relationnelle comme clé de partition dans la rubrique Timestream, et crée un attribut pour chaque colonne de la base de données source. Lorsque vous utilisez `map-record-to-record`, pour toute colonne de la table source non répertoriée dans la liste d'attributs `exclude-columns`, AWS DMS crée un attribut correspondant dans la rubrique cible. Cet attribut correspondant est créé que la colonne source soit ou non utilisée dans un mappage d'attribut.

Une manière de comprendre `map-record-to-record` est de le voir en action. Dans cet exemple, imaginons que vous commencez avec une ligne de table d'une base de données relationnelle, présentant la structure et les données suivantes :

FirstName	LastName	StoreId	HomeAddress	HomePhone	WorkAddress	WorkPhone	DateofBirth
Randy	Marsh	5	221B Baker Street	1234567890	31 Spooner Street, Quahog	9876543210	02/29/1988

Pour migrer ces informations à partir d'un schéma nommé `Test` vers une rubrique Timestream, vous créez des règles pour mapper les données à la rubrique cible. La règle suivante illustre ce mappage.

```
{
    "rules": [
        {
```

```
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "rule-action": "include",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            }
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "2",
            "rule-name": "DefaultMapToTimestream",
            "rule-action": "map-record-to-record",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "Customers"
            }
        }
    ]
}
```

L'exemple ci-dessous illustre le format d'enregistrement obtenu avec une rubrique Timestream et une clé de partition (en l'occurrence, `taskResourceId.schemaName.tableName`) à l'aide de notre exemple de données dans la rubrique Timestream cible :

```
  {
     "FirstName": "Randy",
     "LastName": "Marsh",
     "StoreId":  "5",
     "HomeAddress": "221B Baker Street",
     "HomePhone": "1234567890",
     "WorkAddress": "31 Spooner Street, Quahog",
     "WorkPhone": "9876543210",
     "DateOfBirth": "02/29/1988"
  }
```

Limites d'utilisation d'Amazon Timestream comme cible pour AWS Database Migration Service

Les limites suivantes s'appliquent lorsque vous utilisez Amazon Timestream comme cible :

- Dimensions et horodatages : Timestream utilise les dimensions et les horodatages des données sources comme clé primaire composite, et ne vous autorise pas à mettre à jour ces valeurs. Cela signifie que si vous modifiez l'horodatage ou les dimensions d'un enregistrement dans la base de données source, la base de données Timestream essaiera de créer un nouvel enregistrement. Si vous modifiez la dimension ou l'horodatage d'un enregistrement afin de le faire correspondre à celui d'un autre enregistrement existant, il est possible qu'AWS DMS mette à jour les valeurs de l'autre enregistrement au lieu de créer un nouvel enregistrement ou de mettre à jour l'enregistrement correspondant précédent.
- Commandes DDL : la version actuelle d'AWS DMS ne prend en charge que les commandes DDL `CREATE TABLE` et `DROP TABLE`.
- Limites des enregistrements : Timestream impose des limites pour les enregistrements (taille des enregistrements et des mesures, par exemple). Pour plus d'informations, consultez Quotas dans le Guide du développeur Amazon Timestream.
- Suppression des enregistrements et valeurs nulles : Timestream ne prend pas en charge la suppression des enregistrements. Pour prendre en charge la migration des enregistrements supprimés de la source, AWS DMS efface les champs correspondants dans les enregistrements de la base de données Timestream cible. AWS DMS remplace les valeurs des champs de l'enregistrement cible correspondant par 0 pour les champs numériques, null pour les champs de texte et false pour les champs booléens.
- Timestream en tant que cible ne prend pas en charge les sources qui ne sont pas des bases de données relationnelles (SGBDR).
- AWS DMS prend en charge Timestream en tant que cible dans les régions suivantes uniquement :
  - USA Est (Virginie du Nord)
  - USA Est (Ohio)
  - USA Ouest (Oregon)
  - Europe (Irlande)
  - Europe (Francfort)
  - Asie-Pacifique (Sydney)
  - Asia Pacific (Tokyo)
- Timestream en tant que cible ne prend pas en charge la valeur `TRUNCATE_BEFORE_LOAD` pour le paramètre `TargetTablePrepMode`. Nous vous recommandons d'utiliser `DROP_AND_CREATE` pour ce paramètre.

Utilisation d'Amazon RDS pour Db2 et IBM Db2 LUW en tant que cible pour AWS DMS

Vous pouvez migrer des données d'une base de données Db2 LUW vers une base de données Amazon RDS pour Db2 ou une base de données Db2 sur site à l'aide d'AWS Database Migration Service (AWS DMS).

Pour en savoir plus sur les versions de Db2 LUW prises en charge par AWS DMS en tant que cible, consultez Objectifs pour AWS DMS.

Vous pouvez utiliser le protocole SSL pour chiffrer les connexions entre votre point de terminaison Db2 LUW et l'instance de réplication. Pour plus d'informations sur l'utilisation de SSL avec un point de terminaison Db2 LUW, consultez Utilisation du protocole SSL avec AWS Database Migration Service.

Limites d'utilisation de Db2 LUW en tant que cible pour AWS DMS

Les limites suivantes s'appliquent à l'utilisation d'une base de données Db2 LUW en tant que cible pour AWS DMS. Pour connaître les limites d'utilisation de Db2 LUW en tant que source, consultez Limitations liées à l'utilisation de DB2 LUW comme source pour AWS DMS.

- AWS DMS prend uniquement en charge Db2 LUW en tant que cible lorsque la source est Db2 LUW ou Db2 for z/OS.
- L'utilisation de Db2 LUW en tant que cible ne prend pas en charge les réplications en mode LOB complet.
- L'utilisation de DB2 LUW en tant que cible ne prend pas en charge le type de données XML pendant la phase de chargement complet. Il s'agit d'une limite de l'utilitaire IBM dbload. Pour plus d'informations, consultez Utilitaire dbload dans la documentation Serveurs IBM Informix.
- AWS DMS tronque les champs BLOB avec des valeurs correspondant au guillemet double ("). Il s'agit d'une limite de l'utilitaire IBM dbload.

Paramètres de point de terminaison lors de l'utilisation de DB2 LUW en tant que cible pour AWS DMS

Vous pouvez utiliser des paramètres de point de terminaison pour configurer votre base de données Db2 LUW cible comme si vous utilisiez des attributs de connexion supplémentaires. Vous spécifiez les paramètres lorsque vous créez le point de terminaison cible à l'aide de la console AWS DMS ou

à l'aide de la commande `create-endpoint` dans AWS CLI, avec la syntaxe JSON `--ibm-db2-settings '{"EndpointSetting": "value", ...}'`.

Les paramètres de point de terminaison que vous pouvez utiliser avec Db2 LUW en tant que cible sont indiqués dans le tableau suivant.

Nom	Description
`KeepCsvFiles`	Si c'est vrai, AWS DMS enregistre dans la cible DB2 LUW tous les fichiers .csv qui ont été utilisés pour répliquer les données. DMS utilise ces fichiers à des fins d'analyse et de résolution des problèmes.
`LoadTimeout`	Durée (en millisecondes) avant qu'AWS DMS fasse expirer les opérations effectuées par DMS sur la cible Db2. La valeur par défaut est de 1 200 (20 minutes).
`MaxFileSize`	Spécifie la taille maximale (en Ko) des fichiers .csv utilisés pour transférer des données vers Db2 LUW.
`WriteBufferSize`	Taille (en Ko) du tampon d'écriture de fichier en mémoire utilisé lors de la génération des fichiers .csv sur le disque local au niveau de l'instance de réplication DMS. La valeur par défaut est 1 024 (1 Mo).

Configuration de points de terminaison de VPC en tant que points de terminaison sources et cibles AWS DMS

AWS DMS prend en charge les points de terminaison de cloud privé virtuel (VPC) Amazon en tant que sources et cibles. AWS DMS peut se connecter à n'importe quelle base de données source ou cible AWS avec des points de terminaison Amazon VPC, à condition que les routes explicitement définies vers ces bases de données source et cible soient définies dans leur VPC AWS DMS.

En prenant en charge les points de terminaison Amazon VPC, AWS DMS facilite la conservation de la sécurité réseau de bout en bout pour toutes les tâches de réplication sans configuration réseau ni installation supplémentaires. L'utilisation de points de terminaison de VPC pour tous les points de terminaison sources et cibles garantit que l'ensemble de votre trafic restera dans votre VPC et

sous votre contrôle. Les mises à niveau vers AWS DMS versions 3.4.7 et ultérieures nécessitent que vous configuriez AWS DMS pour qu'il utilise des points de terminaison de VPC ou pour qu'il utilise des routes publiques vers tous vos points de terminaison sources et cibles qui interagissent avec les services Amazon Web Services suivants :

- Amazon S3
- Amazon Kinesis
- AWS Secrets Manager
- Amazon DynamoDB
- Amazon Redshift
- Amazon OpenSearch Service

Vous aurez peut-être besoin de points de terminaison de VPC pour prendre en charge AWS DMS à partir de la version 3.4.7, comme décrit ci-dessous.

Qui est affecté lors d'une migration vers AWS DMS version 3.4.7 ou ultérieure ?

Vous êtes affecté si vous utilisez un ou plusieurs des points de terminaison AWS DMS répertoriés précédemment, et si ces points de terminaison ne sont pas routables publiquement ou n'ont pas de points de terminaison de VPC qui leur soient déjà associés.

Qui n'est pas affecté lors d'une migration vers AWS DMS version 3.4.7 ou ultérieure ?

Vous n'êtes pas affecté si :

- vous n'utilisez pas un ou plusieurs des points de terminaison AWS DMS répertoriés précédemment ;
- vous utilisez un ou plusieurs des points de terminaison répertoriés précédemment et qu'ils sont routables publiquement ;
- vous utilisez un ou plusieurs des points de terminaison répertoriés précédemment et que des points de terminaison de VPC leur sont associés.

Préparation d'une migration vers AWS DMS version 3.4.7 ou ultérieure

Pour éviter l'échec des tâches AWS DMS lorsque vous utilisez l'un des points de terminaison décrits précédemment, effectuez l'une des étapes suivantes avant de mettre à niveau AWS DMS vers la version 3.4.7 ou ultérieure :

- Configurez les points de terminaison AWS DMS affectés pour qu'ils soient routables publiquement. Par exemple, ajoutez une route de passerelle Internet (IGW) à n'importe quel VPC déjà utilisé par votre instance de réplication AWS DMS pour rendre tous ses points de terminaison sources et cibles routables publiquement.
- Créez des points de terminaison de VPC pour accéder à tous les points de terminaison sources et cibles utilisés par AWS DMS, comme décrit ci-dessous.

Pour tous les points de terminaison de VPC existants que vous utilisez pour vos points de terminaison sources et cibles AWS DMS, veillez à ce qu'ils utilisent une politique d'approbation conforme au document de politique XML, `dms-vpc-role` Pour plus d'informations sur ce document de politique XML, consultez Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS.

Sinon, configurez vos instances de réplication en tant que points de terminaison de VPC en ajoutant un point de terminaison de VPC au VPC où elles se trouvent. Si vous avez configuré vos instances de réplication sans points de terminaison publics, l'ajout d'un point de terminaison de VPC publiquement accessible au VPC qui contient vos instances de réplication les rend publiquement accessibles. Vous n'avez rien d'autre à faire pour associer spécifiquement vos instances de réplication au point de terminaison de VPC.

> **Note**
>
> Des services différents peuvent avoir des configurations de point de terminaison de VPC uniques. Par exemple, lors de l'utilisation d'AWS Secrets Manager, vous n'avez généralement pas besoin d'ajuster la table de routage. Vérifiez toujours les exigences spécifiques à chaque service.

Création d'un point de terminaison de VPC sur le VPC contenant votre instance de réplication

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc/.

2. Dans la barre de menu de la console VPC, choisissez la Région AWS de votre instance de réplication AWS DMS.
3. Dans le volet de navigation du VPC, choisissez Points de terminaison.
4. Dans Points de terminaison, choisissez Créer un point de terminaison.
5. Vous pouvez éventuellement spécifier une balise de nom. Par exemple, **`my-endpoint-DynamoDB-01`**.
6. Sous Services pour S3 ou DynamoDB uniquement, choisissez un nom de service dont le type est défini sur Passerelle.
7. Sous VPC, choisissez le même VPC que notre instance de réplication AWS DMS pour créer le point de terminaison.
8. Sous Tables de routage, choisissez toutes les valeurs d'ID de table de routage disponibles.
9. Pour spécifier le contrôle d'accès, sous Politique, choisissez Accès complet. Si vous souhaitez utiliser un outil de création de politiques pour spécifier votre propre contrôle d'accès, choisissez Personnalisé. Dans tous les cas, utilisez une politique d'approbation conforme au document de politique JSON, `dms-vpc-role`. Pour plus d'informations sur ce document de politique, consultez Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS.
10. Sous Points de terminaison, vérifiez que le Statut du point de terminaison de VPC nouvellement créé est Disponible.

Pour plus d'informations sur la configuration des points de terminaison de VPC pour une instance de réplication AWS DMS, consultez Configurations réseau pour la migration de base de données. Pour plus d'informations sur la création de points de terminaison de VPC d'interface pour accéder aux services AWS en général, consultez Accès à un service AWS à l'aide d'un point de terminaison de VPC d'interface dans le Guide AWS PrivateLink. Pour en savoir plus sur la disponibilité régionale d'AWS DMS pour les points de terminaison de VPC, consultez le tableau des régions AWS.

Instructions DDL prises en charge par AWS DMS

Vous pouvez exécuter des instructions de langage de définition de données (Data Definition Language, DDL) sur la base de données source au cours du processus de migration des données. Ces instructions sont répliquées dans la base de données cible par le serveur de réplication.

Les instructions DDL prises en charge sont les suivantes :

- Create table

- Drop table
- Rename table
- Truncate table
- Add column
- Drop column
- Rename column
- Modifier le type de données d'une colonne

DMS ne capture pas toutes les instructions DDL prises en charge pour certains types de moteurs sources. De plus, DMS gère différemment les instructions DDL lorsqu'il les applique à des moteurs cibles spécifiques. Pour en savoir plus sur les instructions DDL prises en charge pour une source spécifique et sur la manière dont elles sont appliquées à une cible, consultez la rubrique de documentation spécifique à ce point de terminaison source et cible.

Vous pouvez utiliser les paramètres de tâche pour configurer la façon dont DMS gère le comportement DDL lors de la capture des données de modification (CDC). Pour plus d'informations, consultez Paramètres de tâche pour la gestion du DDL de traitement des modifications.

Utilisation de tâches AWS DMS

C'est dans le cadre d'une tâche AWS Database Migration Service (AWS DMS) que tout le travail est effectué. Vous spécifiez les tables (ou les vues) et les schémas à utiliser pour la migration, ainsi que tout traitement particulier, tel que les exigences de journalisation, les données des tables de contrôle et la gestion des erreurs.

Une tâche se compose de trois phases principales :

- Migration des données existantes (chargement complet)
- l'application des modifications mises en cache ;
- Réplication continue (capture des données de modification)

Pour plus d'informations et une vue d'ensemble de la façon dont les tâches de migration AWS DMS migrent les données, consultez Vue de haut niveau de AWS DMS.

Lors de la création d'une tâche de migration, il y a plusieurs choses que vous devez connaître :

- Pour pouvoir créer une tâche, veillez à créer un point de terminaison source, un point de terminaison cible et une instance de réplication.
- Vous pouvez spécifier plusieurs paramètres de tâche pour personnaliser votre tâche de migration. Vous pouvez les définir en utilisant la AWS Management Console, l'AWS Command Line Interface (AWS CLI) ou l'API AWS DMS. Il s'agit, entre autres, de spécifier la façon dont les erreurs de migration sont gérées, la journalisation et les informations sur la table de contrôle. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.
- Après avoir créé une tâche, vous pouvez l'exécuter immédiatement. Les tables cibles avec les définitions de métadonnées nécessaires sont automatiquement créées et chargées, et vous pouvez spécifier la réplication continue.
- Par défaut AWS DMS commence votre tâche dès que vous la créez. Toutefois, dans certains cas, vous pouvez reporter le début de la tâche. Par exemple, lorsque vous utilisez l'AWS CLI, vous pouvez avoir un processus qui crée une tâche et un processus différent qui commence la tâche en fonction d'un événement déclencheur. Selon les besoins, vous pouvez reporter le début de la tâche.

- Vous pouvez surveiller, arrêter ou redémarrer des tâches à l'aide de la console, de l'AWS CLI ou de l'API AWS DMS. Pour en savoir plus sur l'arrêt d'une tâche à l'aide de l'API AWS DMS, consultez StopReplicationTask dans Référence d'API AWS DMS.

Voici les actions que vous pouvez entreprendre lorsque vous utilisez une tâche AWS DMS.

Tâche	Documentation utile
Création d'une tâche Lorsque vous créez une tâche, vous spécifiez la source, la cible et l'instance de réplication, ainsi que les paramètres de migration.	Création d'une tâche
Création d'une tâche de réplication continue Vous pouvez configurer une tâche afin de fournir une réplication continue entre la source et la cible.	Création de tâches pour la réplication continue à l'aide d'AWS DMS
Application des paramètres de tâche Chaque tâche possède des paramètres que vous pouvez configurer selon les besoins de votre migration de base de données. Vous créez ces paramètres dans un fichier JSON ou, pour certains paramètres, vous pouvez spécifier les paramètre s à l'aide de la console AWS DMS. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.	Spécification des paramètres des tâches pour les tâches du AWS Database Migration Service

Tâche	Documentation utile
Utilisation d'un mappage de tables Un mappage de tables spécifie des paramètres de tâche supplémentaires pour les tables à l'aide de plusieurs types de règles. Ces règles vous permettent de spécifier la source de données, le schéma source, les tables et les vues, les données, les éventuelles transformations de données et de table devant se produire au cours de la tâche, ainsi que les paramètres spécifiant la manière dont ces tables et colonnes sont migrées de la source vers la cible.	Règles de sélection Règles et actions de sélection Règles de transformation Règles et actions de transformation Règles relatives aux paramètres des tables Règles des paramètres de table et de collection et opérations

Tâche	Documentation utile
Exécution d'évaluations des tâches de prémigration Vous pouvez activer et exécuter des évaluations de tâches de prémigration pour détecter des problèmes avec une base de données source et cible prise en charge, qui peuvent causer des problèmes lors d'une migration. Cela peut inclure des problèmes tels que des types de données non pris en charge, des index et des clés primaires incompatibles, ainsi que d'autres paramètres de tâche en conflit. Ces évaluations de prémigration sont exécutées avant que vous n'exécutiez la tâche, afin d'identifier les problèmes potentiels avant qu'ils ne surviennent pendant une migration.	Activation et utilisation des évaluations de prémigration pour une tâche
Validation des données La validation des données est un paramètre de tâche que vous pouvez utiliser pour qu'AWS DMS compare les données de votre magasin de données cible avec les données de votre magasin de données source.	Validation des données AWS DMS.
Modification d'une tâche Lorsqu'une tâche est arrêtée, vous pouvez modifier les paramètres de la tâche.	Modification d'une tâche

Tâche	Documentation utile
Déplacement d'une tâche Lorsqu'une tâche est arrêtée, vous pouvez la déplacer vers une autre instance de réplication.	Déplacement d'une tâche
Rechargement de tables pendant une tâche Vous pouvez recharger une table pendant une tâche si une erreur se produit pendant la tâche.	Rechargement de tables pendant une tâche
Application de filtres Vous pouvez utiliser des filtres de source pour limiter le nombre et le type d'enregistrements transféré s de votre source vers votre cible. Par exemple, vous pouvez spécifier que seuls les employés avec un emplacement de siège sont déplacés vers la base de données cible. Vous appliquez des filtres sur une colonne de données.	Utilisation de filtres de source
Surveillance d'une tâche Il existe plusieurs façons d'obtenir des informations sur les performan ces d'une tâche et sur les tables utilisées par la tâche.	Surveillance des tâches AWS DMS

Tâche	Documentation utile
Gestion des journaux de tâches Vous pouvez consulter et supprimer des journaux de tâches à l'aide de l'API AWS DMS ou de l'AWS CLI.	Affichage et gestion des journaux de tâches AWS DMS

Rubriques

- Création d'une tâche
- Création de tâches pour la réplication continue à l'aide d'AWS DMS
- Modification d'une tâche
- Déplacement d'une tâche
- Rechargement de tables pendant une tâche
- Utilisation du mappage de table pour spécifier des paramètres de tâche
- Utilisation de filtres de source
- Activation et utilisation des évaluations de prémigration pour une tâche
- Spécification de données supplémentaires pour les paramètres de tâche

Création d'une tâche

Pour créer une tâche de AWS DMS migration, procédez comme suit :

- Créez un point de terminaison source, un point de terminaison cible et une instance de réplication avant de pouvoir créer une tâche de migration.
- Choisissez une méthode de migration :
  - Migration de données vers la base de données cible : ce processus crée des fichiers ou tables dans la base de données cible et définit automatiquement les métadonnées nécessaires au niveau de la cible. Il remplit également les tables avec des données issues de la source. Les données des tables sont chargées en parallèle pour une efficacité améliorée. Ce processus est l'option Migrer les données existantes dans AWS Management Console et est appelée `Full Load` dans l'API.
  - Capture des modifications lors de la migration : ce processus capture les modifications apportées à la base de données source qui se produisent pendant que les données sont migrées de la

source vers la cible. Une fois la migration des données initialement demandées terminée, le processus de capture des données modifiées (CDC) applique les modifications capturées à la base de données cible. Les changements sont capturés et appliqués en tant qu'unités de transactions validées uniques et vous pouvez mettre à jour plusieurs tables cibles différents dans le cadre d'une opération de validation de source unique. Cette approche garantit l'intégrité transactionnelle dans la base de données cible. Ce processus est l'option Migrer les données existantes et répliquer les modifications en cours dans la console et est appelé `full-load-and-cdc` dans l'API.

- Réplication des modifications des données uniquement dans la base de données source : ce processus lit le fichier journal de récupération du système de gestion de base de données source (SGBD) et regroupe les entrées pour chaque transaction. Dans certains cas, AWS DMS impossible d'appliquer les modifications à la cible dans un délai raisonnable (par exemple, si la cible n'est pas accessible). Dans ces cas, les modifications sont mises en AWS DMS mémoire tampon sur le serveur de réplication aussi longtemps que nécessaire. Il ne lit pas à nouveau les journaux du SGBD source, ce qui peut prendre du temps. Ce processus correspond à l'option Répliquer les modifications de données uniquement de la console AWS DMS .

- Déterminez la façon dont la tâche doit gérer les LOB (Large Binary Objects) sur la source. Pour plus d'informations, consultez Configuration du support LOB pour les bases de données sources dans une tâche AWS DMS.
- Spécifiez les paramètres de la tâche de migration. Il s'agit, entre autres, de la configuration de la journalisation, de la spécification des données qui sont écrites dans la table de contrôle de la migration, de la manière dont les erreurs sont gérées et d'autres paramètres. Pour plus d'informations sur les paramètres de tâche, consultez Spécification des paramètres des tâches pour les tâches du AWS Database Migration Service.
- Configurez le mappage de table pour définir les règles de sélection et de filtrage des données que vous migrez. Pour plus d'informations sur le mappage de table, consultez la page Utilisation du mappage de table pour spécifier des paramètres de tâche. Avant de spécifier votre mappage, consultez la section de documentation relative au mappage de type de données pour votre base de données source et cible.
- Activez et exécutez des évaluations de tâche de prémigration avant d'exécuter la tâche. Pour plus d'informations sur les évaluations de prémigration, consultez Activation et utilisation des évaluations de prémigration pour une tâche.
- Spécifiez les données supplémentaires requises pour la tâche de migration de vos données. Pour plus d'informations, consultez Spécification de données supplémentaires pour les paramètres de tâche.

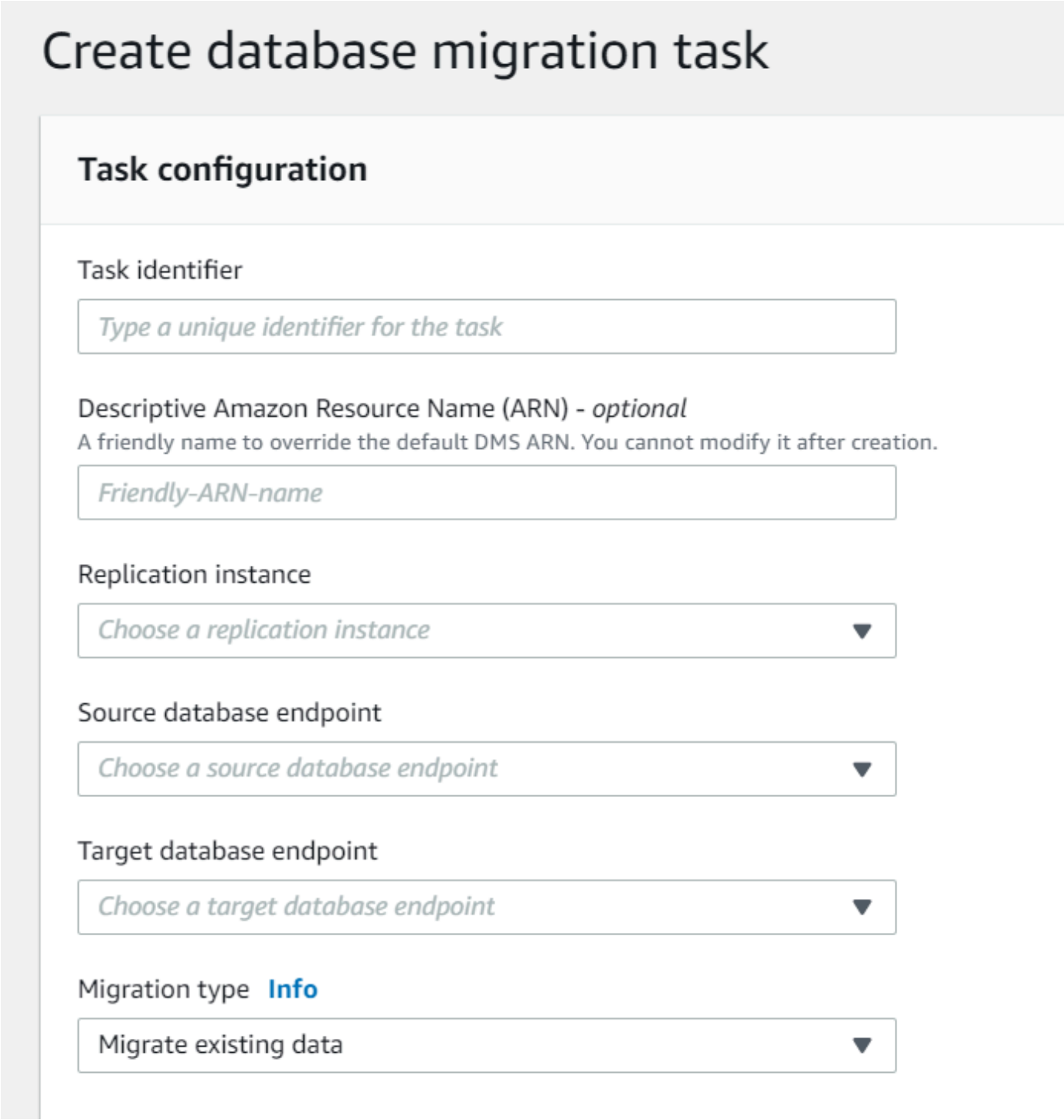
Vous pouvez choisir de démarrer une tâche dès que vous terminez de spécifier les informations pour cette tâche sur la page Créer une tâche. Vous pouvez également démarrer la tâche ultérieurement à partir de la page Tableau de bord.

La procédure suivante suppose que vous avez déjà spécifié des informations d'instance de réplication et des points de terminaison. Pour plus d'informations sur la configuration des points de terminaison, consultez Création de points de terminaison source et cible.

Pour créer une tâche de migration

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.

   Si vous êtes connecté en tant qu'utilisateur AWS Identity and Access Management (IAM), assurez-vous de disposer des autorisations d'accès AWS DMS appropriées. Pour plus d'informations sur les autorisations requises, consultez Autorisations IAM nécessaires pour utiliser AWS DMS.
2. Dans le volet de navigation, choisissez Tâches de migration de base de données, puis choisissez Créer une tâche.
3. Sur la page Créer une tâche de migration de base de données, dans la section Configuration de tâche, spécifiez les options de tâche. Le tableau suivant décrit les paramètres.



Pour cette option	Faire ceci
Identifiant de tâche	Entrez un nom pour la tâche.

Pour cette option	Faire ceci
Amazon Resource Name (ARN) descriptif - facultatif	Un nom convivial pour remplacer l' AWS DMS ARN par défaut. Vous ne pouvez pas modifier ce nom après avoir créé la tâche.
Instance de réplication	Affiche l'instance de réplication à utiliser.
Point de terminaison de base de données source	Affiche le point de terminaison source à utiliser.
Point de terminaison de base de données cible	Affiche le point de terminaison cible à utiliser.
Type de migration	Sélectionnez la méthode de migration que vous souhaitez utiliser. Vous pouvez choisir de migrer simplement les données existantes vers la base de données cible ou d'envoyer les modifications continues à la base de données cible en plus des données migrées.

4. Dans la section Paramètres de tâche, spécifiez les valeurs pour modifier votre tâche, le mode de préparation de la table cible, la tâche d'arrêt, les paramètres LOB, la validation et la journalisation.

Pour cette option	Faire ceci
Mode de modification	Choisissez d'utiliser l'assistant ou l'éditeur JSON pour spécifier les paramètres de votre tâche. Si vous choisissez Assistant, les options suivantes seront affichées.
Mode de départ CDC pour les transactions sources	Ce paramètre n'est visible que si vous choisissez Répliquer les modifications de données uniquement pour Type de migration dans la section précédente. Désactiver le mode de départ CDC personnalisé : si vous choisissez cette option, vous pouvez démarrer votre tâche soit automatiquement en utilisant l'option

Pour cette option	Faire ceci
	Automatiquement lors de la création ci-dessous, soit manuellement à l'aide de la console. Activer le mode de départ CDC personnalisé : si vous choisissez cette option, vous pouvez spécifier une heure de début UTC personnalisée pour commencer à traiter les modifications.

Pour cette option	Faire ceci
Mode de préparation des tables cibles	Ce paramètre n'est visible que si vous choisissez Migrer les données existantes ou Migrer les données existantes et répliquer les modifications continues pour Type de migration dans la section précédente. Ne rien faire : en mode Ne rien faire, AWS DMS suppose que les tables cibles ont été précréées sur la cible. Si les tables ne sont pas vides, des conflits risquent de se produire lors de la migration des données et entraîner une erreur de tâche DMS. Si la table cible n'existe pas, DMS la crée à votre place. Votre structure de table demeure inchangée et toutes les données existantes restent dans la table. Le mode Ne rien faire convient particulièrement pour les tâches CDC uniquement lorsque les tables cibles ont été renvoyées à partir de la source, et la réplicati on continue est appliquée pour préserver la synchroni sation de la source et de la cible. Pour pré-créer des tables, vous pouvez utiliser AWS Schema Conversion Tool (AWS SCT). Pour plus d'informations, consultez la section Installation AWS SCT. Supprimer les tables sur la cible : en mode Supprimer les tables sur la cible, AWS DMS supprime les tables cibles et les recrée avant de lancer la migration. Cette approche garantit que les tables cibles sont vides au début de la migration. AWS DMS crée uniquemen t les objets nécessaires à la migration efficace des données : tables, clés primaires et, dans certains cas, index uniques. AWS DMS ne crée pas d'index secondaires, de contraintes de clé non primaires ou de valeurs par défaut pour les données de colonne. Si vous effectuez un chargement complet plus une tâche CDC ou CDC uniquement, nous vous recommandons de suspendre la migration à ce stade. Ensuite, créez

Pour cette option	Faire ceci
	des index secondaires prenant en charge le filtrage des instructions de mise à jour et de suppression. Vous devrez peut-être effectuer certaines configurations sur la base de données cible lorsque vous utiliserez le mode Supprimer les tables sur la cible. Par exemple, pour une cible Oracle, AWS DMS impossible de créer un schéma (utilisateur de base de données) pour des raisons de sécurité. Dans ce cas, vous créez au préalable l'utilisateur du schéma afin de AWS DMS pouvoir créer les tables au début de la migration. Pour la plupart des autres types de cibles, AWS DMS crée le schéma et toutes les tables associées avec les paramètres de configuration appropriés. Tronquer : en mode tronquer, AWS DMS tronque toutes les tables cibles avant le début de la migration. Si la table cible n'existe pas, DMS la crée à votre place. La structure de votre table reste inchangée, mais les tables sont tronquées au niveau de la cible. Le mode Tronquer convient particulièrement pour les migrations avec un chargement complet ou un chargement complet + CDC, dans lesquelles le schéma cible a été créé au préalable, avant le démarrage de la migration. Pour créer des tables au préalable, vous pouvez utiliser AWS SCT. Pour plus d'informations, consultez la section Installation AWS SCT. **Note** Si votre cible est MongoDB, le mode Tronquer ne tronque pas les tables au niveau de la cible. Au lieu de cela, il supprime la collection et

Pour cette option	Faire ceci
	perd tous les index. Évitez le mode Tronquer lorsque votre cible est MongoDB.
Arrêter la tâche après la fin du chargement complet	Ce paramètre n'est visible que si vous choisisse z Migrer les données existantes et répliquer les modifications continues pour Type de migration dans la section précédente. Ne pas arrêter : n'arrête pas la tâche, mais applique immédiatement les modifications mises en cache et continue. Arrêter avant d'appliquer les modifications mises en cache : arrête la tâche avant l'application des modificat ions mises en cache. En utilisant cette approche, vous pouvez ajouter des index secondaires qui peuvent accélérer l'application des modifications. Arrêter après l'application des modifications mises en cache : arrête la tâche après l'application des modificat ions mises en cache. En utilisant cette approche, vous pouvez ajouter des clés étrangères si vous utilisez une application transactionnelle.
Inclure les colonnes LOB dans la réplication	Ne pas inclure les colonnes LOB : les colonnes LOB sont exclues de la migration. Mode LOB complet : migrez des LOB complets, quelle que soit leur taille. AWS DMS migre les LOB par morceaux en morceaux contrôlés par le paramètre Max LOB size. Ce mode est plus lent que le mode LOB limité. Mode LOB limité : tronque les objets LOB en fonction de la valeur du paramètre Taille de LOB maximale. Ce mode est plus rapide que le mode LOB complet.

Pour cette option	Faire ceci
Taille de LOB maximale (Ko)	En Mode LOB limité, les colonnes LOB qui dépassent le paramètre Taille maximale de LOB sont tronquées à la valeur spécifiée dans Taille maximale de LOB.
Activer la validation	Active la validation des données afin de vérifier que les données sont migrées de manière exacte de la source vers la cible. Pour plus d'informations, consultez Validation des données AWS DMS.
Activer CloudWatch les journaux	Active la journalisation par Amazon CloudWatch.

5. Dans la section Évaluation de pré-migration, choisissez s'il convient d'exécuter une évaluation de prémigration. Une évaluation de prémigration vous avertit des éventuels problèmes de migration avant de démarrer votre tâche de migration de base de données. Pour plus d'informations, consultez Activation et utilisation des évaluations de prémigration.
6. Dans la section Configuration du démarrage de la tâche de migration, indiquez s'il convient de démarrer automatiquement la tâche après sa création.
7. Dans la section Balises, spécifiez les balises dont vous avez besoin pour organiser votre tâche. Vous pouvez utiliser des balises pour gérer vos rôles et politiques IAM et suivre les coûts liés à DMS. Pour plus d'informations, consultez Étiquetage des ressources.
8. Une fois que vous avez fini avec les paramètres de la tâche, sélectionnez Créer une tâche.

Spécification des paramètres des tâches pour les tâches du AWS Database Migration Service

Chaque tâche possède des paramètres que vous pouvez configurer selon les besoins de votre migration de base de données. Vous créez ces paramètres dans un fichier JSON ou, avec certains paramètres, vous pouvez les définir à l'aide de la AWS DMS console. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

Il existe plusieurs types principaux de paramètres de tâche, comme présenté ci-dessous.

Rubriques

- Exemple de paramètres de tâche

- Paramètres de métadonnées des tâches cibles
- Paramètres de tâche de chargement complet
- Paramètres de tâche de voyage dans le temps
- Paramètres de la tâche de journalisation
- Paramètres de tâche de la table de contrôle
- Paramètres de tâche de mémoire tampon de flux
- Paramètres de réglage du traitement des modifications
- Paramètres de la tâche de validation des données
- Paramètres de tâche pour la gestion du DDL de traitement des modifications
- Paramètres de tâche de substitution de caractères
- Paramètres de tâche de l'image antérieure
- Paramètres de traitement des erreurs pour une tâche
- Paramètres de tâche d'enregistrement

Paramètres de tâche	Documentation utile
Création d'un rapport d'évaluation de tâche Vous pouvez créer un rapport d'évaluation de tâche qui présente tous les types de données non pris en charge qui peuvent entraîner des problèmes lors de la migration . Vous pouvez exécuter ce rapport sur votre tâche avant l'exécution de la tâche afin de découvrir les problèmes potentiels.	Activation et utilisation des évaluations de prémigration pour une tâche
Création d'une tâche Lorsque vous créez une tâche, vous spécifiez la source, la cible et	Création d'une tâche

Paramètres de tâche	Documentation utile
l'instance de réplication, ainsi que les paramètres de migration.	
Création d'une tâche de réplication continue Vous pouvez configurer une tâche afin de fournir une réplication continue entre la source et la cible.	Création de tâches pour la réplication continue à l'aide d'AWS DMS
Application des paramètres de tâche Chaque tâche possède des paramètres que vous pouvez configurer selon les besoins de votre migration de base de données. Vous créez ces paramètres dans un fichier JSON ou, avec certains paramètres, vous pouvez les définir à l'aide de la AWS DMS console.	Spécification des paramètres des tâches pour les tâches du AWS Database Migration Service
Validation des données Utilisez la validation des données pour AWS DMS comparer les données de votre magasin de données cible avec celles de votre magasin de données source.	Validation des données AWS DMS
Modification d'une tâche Lorsqu'une tâche est arrêtée, vous pouvez modifier les paramètres de la tâche.	Modification d'une tâche

Paramètres de tâche	Documentation utile
Rechargement de tables pendant une tâche Vous pouvez recharger une table pendant une tâche si une erreur se produit pendant la tâche.	Rechargement de tables pendant une tâche
Utilisation d'un mappage de tables Le mappage de table utilise différent s types de règles afin de spécifier les paramètres de tâche pour la source de données, le schéma source, les données et toutes les transformations qui doivent se produire pendant la tâche.	Règles de sélection Règles et actions de sélection Règles de transformation Règles et actions de transformation
Application de filtres Vous pouvez utiliser des filtres de source pour limiter le nombre et le type d'enregistrements transféré s de votre source vers votre cible. Par exemple, vous pouvez spécifier que seuls les employés avec un emplacement de siège sont déplacés vers la base de données cible. Vous appliquez des filtres sur une colonne de données.	Utilisation de filtres de source
Surveillance d'une tâche Il existe plusieurs façons d'obtenir des informations sur les performan ces d'une tâche et sur les tables utilisées par la tâche.	Surveillance des tâches AWS DMS

Paramètres de tâche	Documentation utile
Gestion des journaux de tâches Vous pouvez consulter et supprimer les journaux des tâches à l'aide de l' AWS DMS API ou AWS CLI.	Affichage et gestion des journaux de tâches AWS DMS

Exemple de paramètres de tâche

Vous pouvez utiliser le AWS Management Console ou le AWS CLI pour créer une tâche de réplication. Si vous utilisez le AWS CLI, vous définissez les paramètres des tâches en créant un fichier JSON, puis en spécifiant l'URI file ://du fichier JSON comme ReplicationTaskSettingsparamètre de l'CreateReplicationTaskopération.

L'exemple suivant montre comment utiliser le AWS CLI pour appeler l'`CreateReplicationTask`opération :

```
aws dms create-replication-task \
--replication-task-identifier MyTask \
--source-endpoint-arn arn:aws:dms:us-
west-2:123456789012:endpoint:ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890ABC \
--target-endpoint-arn arn:aws:dms:us-
west-2:123456789012:endpoint:ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890ABC \
--replication-instance-arn arn:aws:dms:us-
west-2:123456789012:rep:ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890ABC \
--migration-type cdc \
--table-mappings file://tablemappings.json \
--replication-task-settings file://settings.json
```

L'exemple précédent utilise un fichier de mappage de table appelé `tablemappings.json`. Pour obtenir des exemples de mappage de table, consultez Utilisation du mappage de table pour spécifier des paramètres de tâche.

Un fichier JSON de paramètres de tâche peut se présenter comme suit :

```
{
  "TargetMetadata": {
```

```
    "TargetSchema": "",
    "SupportLobs": true,
    "FullLobMode": false,
    "LobChunkSize": 64,
    "LimitedSizeLobMode": true,
    "LobMaxSize": 32,
    "InlineLobMaxSize": 0,
    "LoadMaxFileSize": 0,
    "ParallelLoadThreads": 0,
    "ParallelLoadBufferSize":0,
    "ParallelLoadQueuesPerThread": 1,
    "ParallelApplyThreads": 0,
    "ParallelApplyBufferSize": 100,
    "ParallelApplyQueuesPerThread": 1,
    "BatchApplyEnabled": false,
    "TaskRecoveryTableEnabled": false
  },
  "FullLoadSettings": {
    "TargetTablePrepMode": "DO_NOTHING",
    "CreatePkAfterFullLoad": false,
    "StopTaskCachedChangesApplied": false,
    "StopTaskCachedChangesNotApplied": false,
    "MaxFullLoadSubTasks": 8,
    "TransactionConsistencyTimeout": 600,
    "CommitRate": 10000
  },
    "TTSettings" : {
    "EnableTT" : true,
    "TTS3Settings": {
        "EncryptionMode": "SSE_KMS",
        "ServerSideEncryptionKmsKeyId": "arn:aws:kms:us-west-2:112233445566:key/
myKMSKey",
        "ServiceAccessRoleArn": "arn:aws:iam::112233445566:role/dms-tt-s3-access-role",
        "BucketName": "myttbucket",
        "BucketFolder": "myttfolder",
        "EnableDeletingFromS3OnTaskDelete": false
      },
    "TTRecordSettings": {
        "EnableRawData" : true,
        "OperationsToLog": "DELETE,UPDATE",
        "MaxRecordSize": 64
      }
  },
  "Logging": {
```

```
    "EnableLogging": false
  },
  "ControlTablesSettings": {
    "ControlSchema":"",
    "HistoryTimeslotInMinutes":5,
    "HistoryTableEnabled": false,
    "SuspendedTablesTableEnabled": false,
    "StatusTableEnabled": false
  },
  "StreamBufferSettings": {
    "StreamBufferCount": 3,
    "StreamBufferSizeInMB": 8
  },
  "ChangeProcessingTuning": {
    "BatchApplyPreserveTransaction": true,
    "BatchApplyTimeoutMin": 1,
    "BatchApplyTimeoutMax": 30,
    "BatchApplyMemoryLimit": 500,
    "BatchSplitSize": 0,
    "MinTransactionSize": 1000,
    "CommitTimeout": 1,
    "MemoryLimitTotal": 1024,
    "MemoryKeepTime": 60,
    "StatementCacheSize": 50
  },
  "ChangeProcessingDdlHandlingPolicy": {
    "HandleSourceTableDropped": true,
    "HandleSourceTableTruncated": true,
    "HandleSourceTableAltered": true
  },
  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": true,
    "SourceSchema": "LOOP-DATA",
    "TargetSchema": "loop-data"
  },

  "CharacterSetSettings": {
    "CharacterReplacements": [ {
        "SourceCharacterCodePoint": 35,
        "TargetCharacterCodePoint": 52
      }, {
        "SourceCharacterCodePoint": 37,
        "TargetCharacterCodePoint": 103
      }
```

```
    ],
    "CharacterSetSupport": {
      "CharacterSet": "UTF16_PlatformEndian",
      "ReplaceWithCharacterCodePoint": 0
    }
  },
  "BeforeImageSettings": {
    "EnableBeforeImage": false,
    "FieldName": "",
    "ColumnFilter": "pk-only"
  },
  "ErrorBehavior": {
    "DataErrorPolicy": "LOG_ERROR",
    "DataTruncationErrorPolicy":"LOG_ERROR",
    "DataErrorEscalationPolicy":"SUSPEND_TABLE",
    "DataErrorEscalationCount": 50,
    "TableErrorPolicy":"SUSPEND_TABLE",
    "TableErrorEscalationPolicy":"STOP_TASK",
    "TableErrorEscalationCount": 50,
    "RecoverableErrorCount": 0,
    "RecoverableErrorInterval": 5,
    "RecoverableErrorThrottling": true,
    "RecoverableErrorThrottlingMax": 1800,
    "ApplyErrorDeletePolicy":"IGNORE_RECORD",
    "ApplyErrorInsertPolicy":"LOG_ERROR",
    "ApplyErrorUpdatePolicy":"LOG_ERROR",
    "ApplyErrorEscalationPolicy":"LOG_ERROR",
    "ApplyErrorEscalationCount": 0,
    "FullLoadIgnoreConflicts": true
  },
  "ValidationSettings": {
    "EnableValidation": false,
    "ValidationMode": "ROW_LEVEL",
    "ThreadCount": 5,
    "PartitionSize": 10000,
    "FailureMaxCount": 1000,
    "RecordFailureDelayInMinutes": 5,
    "RecordSuspendDelayInMinutes": 30,
    "MaxKeyColumnSize": 8096,
    "TableFailureMaxCount": 10000,
    "ValidationOnly": false,
    "HandleCollationDiff": false,
    "RecordFailureDelayLimitInMinutes": 1,
    "SkipLobColumns": false,
```

```
    "ValidationPartialLobSize": 0,
    "ValidationQueryCdcDelaySeconds": 0
  }
}
```

Paramètres de métadonnées des tâches cibles

Les paramètres de métadonnées cibles sont les suivants. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

- `TargetSchema` : nom du schéma de la table cible. Si cette option de métadonnées est vide, le schéma de la table source est utilisé. AWS DMS ajoute automatiquement le préfixe du propriétaire de la base de données cible à toutes les tables si aucun schéma source n'est défini. Cette option doit être vide pour les points de terminaison cibles MySQL-type. Le changement de nom d'un schéma dans le mappage de données est prioritaire sur ce paramètre.
- Paramètres LOB : paramètres qui déterminent la façon dont les objets de grande taille (LOB) sont gérés. Si vous définissez `SupportLobs=true`, vous devez définir les éléments suivants à la valeur `true` :
  - `FullLobMode` : si vous définissez cette option sur `true`, vous devez entrer une valeur pour l'option `LobChunkSize`. Entrez la taille, en kilo-octets, des blocs du LOB à utiliser lors de la réplication des données à la cible. L'option `FullLobMode` est la mieux adaptée pour les très grandes tailles de LOB, mais elle a tendance à ralentir le chargement. La valeur recommandée pour `LobChunkSize` est de 64 kilo-octets. L'augmentation de la valeur de `LobChunkSize` au-delà de 64 kilo-octets peut entraîner l'échec des tâches.
  - `InlineLobMaxSize`— Cette valeur détermine les LOB AWS DMS transférés en ligne lors d'un chargement complet. Le transfert de petits LOB est plus efficace que de les rechercher dans une table source. Lors d'un chargement complet, AWS DMS vérifie tous les LOB et effectue un transfert en ligne pour les LOB inférieurs à. `InlineLobMaxSize` AWS DMS transfère tous les LOB supérieurs à l'entrée`InlineLobMaxSize`. `FullLobMode` La valeur par défaut pour `InlineLobMaxSize` est 0 et la plage est de 1 à 102 400 kilo-octets (100 Mo). Définissez une valeur pour `InlineLobMaxSize` uniquement si vous savez que la plupart des LOB sont inférieurs à la valeur spécifiée dans `InlineLobMaxSize`.
  - `LimitedSizeLobMode` : si vous définissez cette option sur `true`, vous devez entrer une valeur pour l'option `LobMaxSize`. Saisissez la taille maximale, en kilo-octets, pour un LOB individuel. La valeur maximale recommandée pour `LobMaxSize` est de 102 400 kilo-octets (100 Mo).

Pour plus d'informations sur les critères d'utilisation de ces paramètres LOB de tâche, consultez Configuration du support LOB pour les bases de données sources dans une tâche AWS DMS. Vous pouvez également contrôler la gestion des LOB pour les tables individuelles. Pour plus d'informations, consultez Règles des paramètres de table et de collection et opérations.

- `LoadMaxFileSize` : option pour les points de terminaison cibles basés sur CSV tels que MySQL, PostgreSQL et Amazon Redshift, qui prend en charge l'utilisation de fichiers .csv (valeurs séparées par des virgules) pour le chargement des données. `LoadMaxFileSize` définit la taille maximale sur le disque des données stockées et déchargées, telles que les fichiers .csv. Cette option remplace l'attribut de connexion de point de terminaison cible `maxFileSize`. Vous pouvez fournir des valeurs 0, qui indiquent que cette option ne remplace pas l'attribut de connexion, à 100 000 Ko.
- `BatchApplyEnabled` : détermine si chaque transaction est appliquée individuellement ou si les modifications sont validées par lots. La valeur par défaut est `false`.

  Lorsque `BatchApplyEnabled` est défini sur `true`, DMS nécessite une clé primaire (PK) ou une clé unique (UK) sur la ou les tables sources. Sans PK ou UK dans les tables sources, seules les insertions par lots sont appliquées, pas les mises à jour ni les suppressions par lots.

  Lorsque `BatchApplyEnabled` est défini sur `true`, AWS DMS génère un message d'erreur si une table cible possède une contrainte unique et une clé primaire. Les tables cibles comportant à la fois une contrainte unique et une clé primaire ne sont pas prises en charge lorsque `BatchApplyEnabled` est défini sur `true`.

  Lorsqu'elle `BatchApplyEnabled` est définie sur true et qu'elle AWS DMS rencontre une erreur de données dans une table dotée de la politique de gestion des erreurs par défaut, la AWS DMS tâche passe du mode batch au one-by-one mode pour les autres tables. Pour modifier ce comportement, vous pouvez définir l'action `"SUSPEND_TABLE"` sur les politiques suivantes dans la propriété de groupe `"ErrorBehavior"` du fichier JSON des paramètres de tâche :

  - `DataErrorPolicy`
  - `ApplyErrorDeletePolicy`
  - `ApplyErrorInsertPolicy`
  - `ApplyErrorUpdatePolicy`

  Pour plus d'informations sur cette propriété de groupe `"ErrorBehavior"`, consultez l'exemple de fichier JSON des paramètres de tâche dans Spécification des paramètres des tâches pour les tâches du AWS Database Migration Service. Une fois ces politiques définies

sur"SUSPEND_TABLE", la AWS DMS tâche suspend les erreurs de données sur toutes les tables qui les génèrent et continue en mode batch pour toutes les tables.

Vous pouvez utiliser le paramètre `BatchApplyEnabled` avec le paramètre `BatchApplyPreserveTransaction`. Si `BatchApplyEnabled` est défini sur `true`, le paramètre `BatchApplyPreserveTransaction` détermine l'intégrité transactionnelle.

Si `BatchApplyPreserveTransaction` est défini sur `true`, l'intégrité transactionnelle est préservée et un lot est assuré de contenir toutes les modifications effectuées dans une transaction à partir de la source.

Si `BatchApplyPreserveTransaction` est défini sur `false`, il peut exister des écarts temporaires dans l'intégrité transactionnelle afin d'améliorer les performances.

Le paramètre `BatchApplyPreserveTransaction` s'applique uniquement aux points de terminaison cible Oracle, et est approprié uniquement lorsque le paramètre `BatchApplyEnabled` est défini sur `true`.

Lorsque les colonnes LOB sont incluses dans la réplication, vous pouvez utiliser `BatchApplyEnabled` uniquement en mode LOB limité.

Pour plus d'informations sur l'utilisation de ces paramètres pour un chargement de capture des données de modification (CDC), consultez Paramètres de réglage du traitement des modifications.

- `MaxFullLoadSubTasks` : indique le nombre maximal de tables à charger en parallèle. La valeur par défaut est 8 ; la valeur maximale 49.
- `ParallelLoadThreads`— Spécifie le nombre de threads AWS DMS utilisés pour charger chaque table dans la base de données cible. Ce paramètre possède des valeurs maximales pour les cibles autres que les SGBDR. La valeur maximale pour une cible DynamoDB est 200. La valeur maximale pour une cible Amazon Kinesis Data Streams, Apache Kafka ou OpenSearch Amazon Service est de 32. Vous pouvez demander une augmentation de cette limite maximale. `ParallelLoadThreads` s'applique aux tâches de chargement complet. Pour plus d'informations sur les paramètres qui activent le chargement parallèle des tables individuelles, consultez Règles des paramètres de table et de collection et opérations.

  Ce paramètre s'applique aux types de moteurs de point de terminaison suivants :

  - DynamoDB
  - Amazon Kinesis Data Streams
  - Amazon MSK

- Amazon OpenSearch Service
- Amazon Redshift

AWS DMS prend en `ParallelLoadThreads` charge MySQL en tant qu'attribut de connexion supplémentaire. `ParallelLoadThreads`ne s'applique pas à MySQL en tant que paramètre de tâche.

- `ParallelLoadBufferSize` : spécifie le nombre maximal d'enregistrements à stocker dans la mémoire tampon utilisée par les threads de chargement parallèles pour charger les données sur la cible. La valeur par défaut est 50. La valeur maximale est 1 000. Ce paramètre n'est actuellement valide que lorsque DynamoDB, Kinesis, Apache Kafka OpenSearch ou R est la cible. Utilisez ce paramètre avec `ParallelLoadThreads`. `ParallelLoadBufferSize` est valide uniquement dans le cas de plusieurs threads. Pour plus d'informations sur les paramètres qui activent le chargement parallèle des tables individuelles, consultez Règles des paramètres de table et de collection et opérations.
- `ParallelLoadQueuesPerThread` : spécifie le nombre de files d'attente auxquelles chaque thread simultané accède pour extraire les enregistrements de données des files d'attente et générer un chargement par lots pour la cible. La valeur par défaut est 1. Ce paramètre n'est actuellement valide que lorsque Kinesis ou Apache Kafka est la cible.
- `ParallelApplyThreads`— Spécifie le nombre de threads simultanés AWS DMS utilisés lors d'un chargement par le CDC pour transférer des enregistrements de données vers un point de terminaison cible Amazon DocumentDB, Kinesis, Amazon MSK OpenSearch ou Amazon Redshift. La valeur par défaut est zéro (0).

  Ce paramètre ne s'applique qu'à la CDC uniquement. Ce paramètre ne s'applique pas au chargement complet.

  Ce paramètre s'applique aux types de moteurs de point de terminaison suivants :

  - Amazon DocumentDB (compatible avec MongoDB)
  - Amazon Kinesis Data Streams
  - Amazon Managed Streaming for Apache Kafka
  - Amazon OpenSearch Service
  - Amazon Redshift
- `ParallelApplyBufferSize`— Spécifie le nombre maximum d'enregistrements à stocker dans chaque file d'attente tampon pour les threads simultanés à envoyer vers un point de terminaison cible Amazon DocumentDB, Kinesis, Amazon MSK OpenSearch ou Amazon Redshift lors d'un

chargement CDC. La valeur par défaut est 100. La valeur maximale est de 1 000. Utilisez cette option lorsque `ParallelApplyThreads` spécifie plusieurs threads.

- `ParallelApplyQueuesPerThread`— Spécifie le nombre de files d'attente auxquelles chaque thread accède pour extraire des enregistrements de données des files d'attente et générer un chargement par lots pour un Amazon DocumentDB, Kinesis, Amazon MSK ou un point de terminaison pendant le CDC. OpenSearch La valeur par défaut est 1.

Paramètres de tâche de chargement complet

Les paramètres de chargement complet sont les suivants. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

- Pour indiquer comment gérer le chargement de la cible au démarrage du chargement complet, spécifiez l'une des valeurs suivantes pour l'option `TargetTablePrepMode` :
  - `DO_NOTHING` : les données et les métadonnées de la table cible existante ne sont pas concernées.
  - `DROP_AND_CREATE` : la table existante est supprimée et une nouvelle table est créée à la place.
  - `TRUNCATE_BEFORE_LOAD` : les données sont tronquées sans impact sur les métadonnées de la table.
- Pour retarder la création d'une clé primaire ou d'un index unique jusqu'à la fin d'un chargement complet, définissez l'option `CreatePkAfterFullLoad` sur `true`.
- Pour les tâches de chargement complet et CDC, vous pouvez définir les options suivantes pour `Stop task after full load completes` :
  - `StopTaskCachedChangesApplied` : définissez cette option sur `true` pour arrêter une tâche après la fin d'un chargement complet et pour appliquer les modifications mises en cache.
  - `StopTaskCachedChangesNotApplied` : définissez cette option sur `true` pour arrêter une tâche avant l'application des modifications mises en cache.
- Pour indiquer le nombre maximal de tables à charger en parallèle, définissez l'option `MaxFullLoadSubTasks`. La valeur par défaut est 8 ; la valeur maximale 49.
- Définissez l'option `ParallelLoadThreads` pour indiquer le nombre de threads simultanés que DMS utilisera pendant un processus de chargement complet pour transférer des enregistrements de données vers un point de terminaison cible. La valeur par défaut est zéro (0).

> **Important**
>
> `MaxFullLoadSubTasks` contrôle le nombre de tables ou de segments de table à charger en parallèle. `ParallelLoadThreads` contrôle le nombre de threads utilisés par une tâche de migration pour exécuter les chargements en parallèle. Ces paramètres sont multiplicatifs. Ainsi, le nombre total de threads utilisés lors d'une tâche de chargement complet correspond approximativement au résultat de la valeur de `ParallelLoadThreads` multipliée par la valeur de `MaxFullLoadSubTasks` (`ParallelLoadThreads` * `MaxFullLoadSubtasks`)).
> Si vous créez des tâches avec un grand nombre de sous-tâches de chargement complet et un grand nombre de threads de chargement parallèle, votre tâche peut consommer trop de mémoire et échouer.

- Vous pouvez définir le nombre de secondes qui attendent AWS DMS la clôture des transactions avant de commencer une opération de chargement complet. Pour ce faire, si les transactions sont ouvertes lorsque la tâche démarre, définissez l'option `TransactionConsistencyTimeout`. La valeur par défaut est 600 (10 minutes). AWS DMS commence le chargement complet une fois que le délai d'expiration est atteint, même si des transactions sont ouvertes. Une full-load-only tâche n'attend pas 10 minutes mais démarre immédiatement.
- Pour indiquer le nombre maximal d'enregistrements qui peuvent être transférés ensemble, définissez l'option `CommitRate`. La valeur par défaut est 10 000 et la valeur maximale est 50 000.

Paramètres de tâche de voyage dans le temps

Pour enregistrer et déboguer les tâches de réplication, vous pouvez utiliser AWS DMS Time Travel. Dans cette approche, vous utilisez Amazon S3 pour stocker les journaux et les chiffrer à l'aide de vos clés de chiffrement. Ce n'est qu'en accédant à votre compartiment S3 de voyage dans le temps que vous pouvez récupérer vos journaux S3 à l'aide de filtres de date et d'heure, puis afficher, télécharger et masquer les journaux selon vos besoins. Ce faisant, vous pouvez « voyager dans le temps » en toute sécurité pour examiner les activités de base de données. Time Travel fonctionne indépendamment de la CloudWatch journalisation. Pour plus d'informations sur la CloudWatch journalisation, consultezParamètres de la tâche de journalisation.

Vous pouvez utiliser Time Travel dans toutes les AWS régions avec des points de terminaison sources AWS DMS compatibles avec Oracle, Microsoft SQL Server et PostgreSQL et des points de terminaison cibles PostgreSQL et MySQL AWS DMS pris en charge. Vous pouvez activer le voyage

dans le temps uniquement pour les tâches de chargement complet et de capture des données de modification (CDC) et pour les tâches de CDC uniquement. Pour activer le voyage dans le temps ou pour modifier des paramètres de voyage dans le temps existants, assurez-vous que votre tâche de réplication est arrêtée.

Les paramètres de voyage dans le temps incluent les propriétés `TTSettings` suivantes :

- `EnableTT` : si cette option est définie sur `true`, la journalisation de voyage dans le temps est activée pour la tâche. La valeur par défaut est `false`.

  Type : booléen

  Obligatoire : non
- `EncryptionMode` : type de chiffrement côté serveur actuellement utilisé sur votre compartiment S3 pour stocker vos données et vos journaux. Vous pouvez spécifier `"SSE_S3"` (par défaut) ou `"SSE_KMS"`.

  Vous pouvez redéfinir `EncryptionMode` de `"SSE_KMS"` à `"SSE_S3"`, mais pas l'inverse.

  Type : chaîne

  Obligatoire : non
- `ServerSideEncryptionKmsKeyId`— Si vous le spécifiez `"SSE_KMS"` pour`EncryptionMode`, fournissez l'ID de votre AWS KMS clé gérée personnalisée. Assurez-vous que la clé que vous utilisez est associée à une politique qui active les autorisations utilisateur AWS Identity and Access Management (IAM) et autorise l'utilisation de la clé.

  Seule votre propre clé KMS symétrique gérée personnalisée est prise en charge avec l'option `"SSE_KMS"`.

  Type : chaîne

  Obligatoire uniquement si vous définissez `EncryptionMode` sur `"SSE_KMS"`
- `ServiceAccessRoleArn` : Amazon Resource Name (ARN) utilisé par le service pour accéder au rôle IAM. Définissez le nom du rôle sur `dms-tt-s3-access-role`. Il s'agit d'un paramètre obligatoire qui permet d' AWS DMS écrire et de lire des objets à partir d'un compartiment S3.

  Type : chaîne

  Obligatoire si le voyage dans le temps est activé

Voici un exemple de politique pour ce rôle.

```
{
 "Version": "2012-10-17",
 "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "kms:GenerateDataKey",
                "kms:Decrypt",
                "s3:ListBucket",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::S3bucketName*",
                "arn:aws:kms:us-east-1:112233445566:key/1234a1a1-1m2m-1z2z-
d1d2-12dmstt1234"
            ]
        }
    ]
}
```

Voici un exemple de politique d'approbation pour ce rôle.

```
{
 "Version": "2012-10-17",
 "Statement": [
         {
             "Effect": "Allow",
             "Principal": {
                 "Service": [
                     "dms.amazonaws.com"
                 ]
             },
             "Action": "sts:AssumeRole"
        }
    ]
}
```

- `BucketName` : nom du compartiment S3 dans lequel stocker les journaux de voyage dans le temps. Assurez-vous de créer ce compartiment S3 avant d'activer les journaux de voyage dans le temps.

  Type : chaîne

  Obligatoire si le voyage dans le temps est activé

- `BucketFolder` : paramètre facultatif pour définir le nom d'un dossier dans le compartiment S3. Si vous spécifiez ce paramètre, DMS crée les journaux de voyage dans le temps dans le chemin "*/BucketName/BucketFolder/taskARN/YYYY/MM/DD/hh*". Si vous ne spécifiez pas ce paramètre, AWS DMS crée le chemin par défaut sous la forme"*/BucketName*/dms-time-travel-logs/*taskARN*/*YYYY*/*MM*/*DD*/*hh*.

  Type : chaîne

  Obligatoire : non

- `EnableDeletingFromS3OnTaskDelete`— Lorsque cette option est définie sur`true`, AWS DMS supprime les journaux de voyage dans le temps de S3 si la tâche est supprimée. La valeur par défaut est `false`.

  Type : chaîne

  Obligatoire : non

- `EnableRawData` : lorsque cette option est définie sur `true`, les données brutes DML pour les journaux de voyage dans le temps apparaissent sous la colonne `raw_data` des journaux de voyage dans le temps. Pour obtenir des détails, consultez Utilisation des journaux de voyage dans le temps. La valeur par défaut est `false`. Lorsque cette option est définie sur `false`, seul le type de DML est capturé.

  Type : chaîne

  Obligatoire : non

- `RawDataFormat`— Dans AWS DMS les versions 3.5.0 et supérieures, quand `EnableRawData` est défini sur. `true` Cette propriété spécifie un format pour les données brutes du DML dans un journal de voyage dans le temps. Elle peut se présenter comme suit :

  - `"TEXT"` : noms de colonnes et valeurs lisibles et analysés pour les événements DML capturés pendant la CDC sous forme de champs `Raw`.

- `"HEX"` : valeur hexadécimale d'origine pour les noms de colonnes et les valeurs capturés pour les événements DML pendant la CDC.

  Cette propriété s'applique aux sources de base de données Oracle et Microsoft SQL Server.

  Type : chaîne

  Obligatoire : non

- `OperationsToLog` : spécifie le type d'opérations DML pour journaliser les journaux de voyage dans le temps. Vous pouvez spécifier l'une des options suivantes :
  - `"INSERT"`
  - `"UPDATE"`
  - `"DELETE"`
  - `"COMMIT"`
  - `"ROLLBACK"`
  - `"ALL"`

  L'argument par défaut est `"ALL"`.

  Type : chaîne

  Obligatoire : non

- `MaxRecordSize` : spécifie la taille maximale des enregistrements de journalisation de voyage dans le temps consignés pour chaque ligne. Utilisez cette propriété pour contrôler la croissance des journaux de voyage dans le temps pour les tables particulièrement chargées. La valeur par défaut est de 64 Ko.

  Type : entier

  Obligatoire : non

Pour plus d'informations sur l'activation et l'utilisation des journaux de voyage dans le temps, consultez les rubriques suivantes.

Rubriques

- Activation des journaux de voyage dans le temps pour une tâche
- Utilisation des journaux de voyage dans le temps

- À quelle fréquence les journaux de voyage AWS DMS dans le temps sont-ils téléchargés sur S3 ?

Activation des journaux de voyage dans le temps pour une tâche

Vous pouvez activer Time Travel pour une AWS DMS tâche à l'aide des paramètres de tâche décrits précédemment. Assurez-vous que votre tâche de réplication est arrêtée avant d'activer le voyage dans le temps.

Pour activer Time Travel à l'aide du AWS CLI

1. Créez un fichier JSON de configuration de tâche DMS et ajoutez la section `TTSettings` suivante. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

```
 .
 .
 .
    },
"TTSettings" : {
  "EnableTT" : true,
  "TTS3Settings": {
      "EncryptionMode": "SSE_KMS",
      "ServerSideEncryptionKmsKeyId": "arn:aws:kms:us-west-2:112233445566:key/
myKMSKey",
      "ServiceAccessRoleArn": "arn:aws:iam::112233445566:role/dms-tt-s3-access-
role",
      "BucketName": "myttbucket",
      "BucketFolder": "myttfolder",
      "EnableDeletingFromS3OnTaskDelete": false
    },
  "TTRecordSettings": {
      "EnableRawData" : true,
      "OperationsToLog": "DELETE,UPDATE",
      "MaxRecordSize": 64
    },
 .
 .
 .
```

2. Dans une action de tâche appropriée, spécifiez ce fichier JSON à l'aide de l'option `--replication-task-settings`. Par exemple, le fragment de code CLI suivant spécifie

ce fichier de paramètres de voyage dans le temps comme faisant partie de `create-replication-task`.

```
aws dms create-replication-task
--target-endpoint-arn arn:aws:dms:us-
east-1:112233445566:endpoint:ELS5O7YTYV452CAZR2EYBNQGILFHQIFVPWFRQAY \
--source-endpoint-arn arn:aws:dms:us-
east-1:112233445566:endpoint:HNX2BWIIN5ZYFF7F6UFFZVWTDFFSMTNOV2FTXZA \
--replication-instance-arn arn:aws:dms:us-
east-1:112233445566:rep:ERLHG2UA52EEJJKFYNYWRPCG6T7EPUAB5AWBUJQ \
--migration-type full-load-and-cdc --table-mappings 'file:///FilePath/
mappings.json' \
--replication-task-settings 'file:///FilePath/task-settings-tt-enabled.json' \
--replication-task-identifier test-task
                            .
                            .
                            .
```

Ici, le nom de ce fichier de paramètres de voyage dans le temps est `task-settings-tt-enabled.json`.

De même, vous pouvez spécifier ce fichier comme faisant partie de l'action `modify-replication-task`.

Notez le traitement spécial des journaux de voyage dans le temps pour les actions de tâche suivantes :

- `start-replication-task` : lorsque vous exécutez une tâche de réplication, si un compartiment S3 utilisé pour le voyage dans le temps n'est pas accessible, la tâche est marquée comme `FAILED`.
- `stop-replication-task`— Lorsque la tâche s'arrête, AWS DMS envoie immédiatement tous les journaux de voyage dans le temps actuellement disponibles pour l'instance de réplication vers le compartiment S3 utilisé pour Time Travel.

Pendant l'exécution d'une tâche de réplication, vous pouvez modifier la valeur `EncryptionMode` de `"SSE_KMS"` à `"SSE_S3"`, mais pas l'inverse.

Si la taille des journaux de voyage dans le temps pour une tâche continue dépasse 1 Go, DMS transfère les journaux vers S3 dans un délai de cinq minutes après avoir atteint cette taille. Après l'exécution d'une tâche, si le compartiment S3 ou la clé KMS devient inaccessible, DMS arrête de transférer les journaux vers ce compartiment. Si vous constatez que vos journaux ne sont pas transférés vers votre compartiment S3, vérifiez votre S3 et vos AWS KMS autorisations. Pour plus de détails sur la fréquence à laquelle DMS transfère ces journaux vers S3, consultez À quelle fréquence les journaux de voyage AWS DMS dans le temps sont-ils téléchargés sur S3 ?.

Pour activer le voyage dans le temps pour une tâche existante à partir de la console, utilisez l'option de l'éditeur JSON sous Paramètres de tâche pour ajouter une section `TTSettings`.

Utilisation des journaux de voyage dans le temps

Les fichiers journaux de voyage dans le temps sont des fichiers CSV (valeurs séparées par des virgules) comportant les champs suivants.

```
log_timestamp
component
dms_source_code_location
transaction_id
event_id
event_timestamp
lsn/scn
primary_key
record_type
event_type
schema_name
table_name
statement
action
result
raw_data
```

Une fois que vos journaux de voyage dans le temps sont disponibles dans S3, vous pouvez y accéder directement et les interroger à l'aide d'outils tels qu'Amazon Athena. Vous pouvez également télécharger les journaux comme n'importe quel fichier à partir de S3.

L'exemple suivant illustre un journal de voyage dans le temps dans lequel les transactions d'une table appelée `mytable` sont journalisées. Les fins de ligne du journal suivant ont été ajoutées pour plus de lisibilité.

```
"log_timestamp ","tt_record_type","dms_source_code_location ","transaction_id",
"event_id","event_timestamp","scn_lsn","primary_key","record_type","event_type",
"schema_name","table_name","statement","action","result","raw_data"
"2021-09-23T01:03:00:778230","SOURCE_CAPTURE","postgres_endpoint_wal_engine.c:00819",
"609284109","565612992","2021-09-23 01:03:00.765321+00","00000E9C/D53AB518","","DML",
"UPDATE (3)","dmstest","mytable","","Migrate","","table dmstest.mytable:
UPDATE: id[bigint]:2244937 phone_number[character varying]:'phone-number-482'
age[integer]:82 gender[character]:'f' isactive[character]:'true '
date_of_travel[timestamp without time zone]:'2021-09-23 01:03:00.76593'
description[text]:'TEST DATA TEST DATA TEST DATA TEST DATA'"
```

À quelle fréquence les journaux de voyage AWS DMS dans le temps sont-ils téléchargés sur S3 ?

Pour minimiser l'utilisation du stockage par votre instance de réplication, AWS DMS déchargez régulièrement les journaux Time Travel de cette instance.

Les journaux de voyage dans le temps sont transférés vers votre compartiment Amazon S3 dans les cas suivants :

- Si la taille actuelle des journaux dépasse 1 Go, AWS DMS télécharge les journaux vers S3 dans les cinq minutes. Ainsi, AWS DMS vous pouvez effectuer jusqu'à 12 appels par heure à S3 et AWS KMS pour chaque tâche en cours d'exécution.
- AWS DMS télécharge les journaux sur S3 toutes les heures, quelle que soit leur taille.
- Lorsqu'une tâche est arrêtée, télécharge AWS DMS immédiatement les journaux de voyage dans le temps vers S3.

Paramètres de la tâche de journalisation

La journalisation utilise Amazon CloudWatch pour enregistrer les informations pendant le processus de migration. Les paramètres de tâche de journalisation vous permettent de spécifier les activités de composant qui sont consignées et la quantité d'informations écrites dans le journal. Les paramètres de tâche de journalisation sont écrits dans un fichier JSON. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

Vous pouvez activer la CloudWatch journalisation de différentes manières. Vous pouvez sélectionner l'`EnableLogging`option AWS Management Console lorsque vous créez une tâche de migration. Vous pouvez également définir l'`EnableLogging`option sur `true` lors de la création d'une tâche à

l'aide de l' AWS DMS API. Vous pouvez également spécifier `"EnableLogging": true` dans le code JSON de la section de journalisation des paramètres de la tâche.

Lorsque vous définissez sur`true`, `EnableLogging` AWS DMS assignez le nom du CloudWatch groupe et le nom du flux comme suit. Vous ne pouvez pas définir ces valeurs directement.

- CloudWatchLogGroup: `dms-tasks-<REPLICATION_INSTANCE_IDENTIFIER>`
- CloudWatchLogStream: `dms-task-<REPLICATION_TASK_EXTERNAL_RESOURCE_ID>`

`<REPLICATION_INSTANCE_IDENTIFIER>` est l'identifiant de l'instance de réplication. `<REPLICATION_TASK_EXTERNAL_RESOURCE_ID>` est la valeur de la section `<resourcename>` de l'ARN de tâche. Pour plus d'informations sur le mode AWS DMS de génération des ARN de ressources, consultezCréation d'un nom de ressource Amazon (ARN) pour AWS DMS.

CloudWatch s'intègre à AWS Identity and Access Management (IAM), et vous pouvez spécifier CloudWatch les actions qu'un utilisateur de votre AWS compte peut effectuer. Pour plus d'informations sur l'utilisation d'IAM dans CloudWatch, consultez Gestion des identités et des accès pour Amazon CloudWatch et Journalisation des appels d' CloudWatch API Amazon dans le guide de l' CloudWatch utilisateur Amazon.

Pour supprimer les journaux d'une tâche, vous pouvez définir `DeleteTaskLogs` sur true dans le code JSON de la section de journalisation des paramètres de la tâche.

Vous pouvez spécifier la journalisation pour les types d'événements suivants :

- `FILE_FACTORY` : la fabrique de fichiers gère les fichiers utilisés pour l'application et le chargement par lots, et gère les points de terminaison Amazon S3.
- `METADATA_MANAGER` : le gestionnaire de métadonnées gère les métadonnées sources et cibles, le partitionnement et l'état des tables pendant la réplication.
- `SORTER` : `SORTER` reçoit les événements entrants du processus `SOURCE_CAPTURE`. Les événements sont regroupés dans des transactions et transmis au composant de service `TARGET_APPLY`. Si le processus `SOURCE_CAPTURE` produit des événements plus rapidement que le composant `TARGET_APPLY` ne peut les consommer, le composant `SORTER` met en cache les événements en attente sur le disque ou dans un fichier d'échange. Les événements mis en cache sont une cause fréquente de manque d'espace de stockage dans les instances de réplication.

  Le composant de service `SORTER` gère les événements mis en cache, collecte les statistiques de CDC et signale le temps de latence des tâches.

- SOURCE_CAPTURE : les données de réplication continue (CDC) sont capturées à partir de la base de données ou du service source et sont transmises au composant de service SORTER.
- SOURCE_UNLOAD : les données sont déchargées de la base de données ou du service source pendant le chargement complet.
- TABLES_MANAGER : le gestionnaire de tables suit les tables capturées, gère l'ordre de migration des tables et collecte les statistiques des tables.
- TARGET_APPLY : les données et les instructions DDL sont appliquées à la base de données cible.
- TARGET_LOAD : les données sont chargées dans la base de données cible.
- TASK_MANAGER : le gestionnaire de tâches gère les tâches en cours et les décompose en sous-tâches pour le traitement parallèle des données.
- TRANSFORMATION : événements de transformation liés au mappage de table. Pour plus d'informations, consultez Utilisation du mappage de table pour spécifier des paramètres de tâche.
- VALIDATOR/ VALIDATOR_EXT : le composant de service VALIDATOR vérifie que les données ont été migrées correctement de la source vers la cible. Pour plus d'informations, consultez Validation des données.

Les composants de journalisation suivants génèrent un grand nombre de journaux lorsque le niveau de gravité des journaux LOGGER_SEVERITY_DETAILED_DEBUG est utilisé :

- COMMON
- ADDONS
- DATA_STRUCTURE
- COMMUNICATION
- FILE_TRANSFER
- FILE_FACTORY

Les niveaux de journalisation autres que DEFAULT sont rarement nécessaires pour ces composants lors du dépannage. Nous ne recommandons pas de modifier le niveau de journalisation DEFAULT pour ces composants, sauf demande spécifique du AWS Support.

Après avoir spécifié l'une des actions précédentes, vous pouvez spécifier la quantité d'informations consignées, comme indiqué dans la liste suivante.

Les niveaux de gravité sont dans l'ordre du plus bas au plus haut niveau d'information. Les niveaux supérieurs englobent toujours les informations des niveaux inférieurs.

- LOGGER_SEVERITY_ERROR : les messages d'erreur sont écrits dans le journal.
- LOGGER_SEVERITY_WARNING : les avertissements et les messages d'erreur sont écrits dans le journal.
- LOGGER_SEVERITY_INFO : les messages d'information, les avertissements et les messages d'erreur sont écrits dans le journal.
- LOGGER_SEVERITY_DEFAULT : les messages d'information, les avertissements et les messages d'erreur sont écrits dans le journal.
- LOGGER_SEVERITY_DEBUG : les messages de débogage, les messages d'information, les avertissements et les messages d'erreur sont écrits dans le journal.
- LOGGER_SEVERITY_DETAILED_DEBUG : toutes les informations sont écrites dans le journal.

L'exemple JSON suivant montre les paramètres de tâche pour consigner toutes les actions et tous les niveaux de gravité.

```
…
  "Logging": {
    "EnableLogging": true,
    "LogComponents": [
      {
        "Id": "FILE_FACTORY",
        "Severity": "LOGGER_SEVERITY_DEFAULT"
      },{
        "Id": "METADATA_MANAGER",
        "Severity": "LOGGER_SEVERITY_DEFAULT"
      },{
        "Id": "SORTER",
        "Severity": "LOGGER_SEVERITY_DEFAULT"
      },{
        "Id": "SOURCE_CAPTURE",
        "Severity": "LOGGER_SEVERITY_DEFAULT"
      },{
        "Id": "SOURCE_UNLOAD",
        "Severity": "LOGGER_SEVERITY_DEFAULT"
      },{
        "Id": "TABLES_MANAGER",
        "Severity": "LOGGER_SEVERITY_DEFAULT"
      },{
        "Id": "TARGET_APPLY",
        "Severity": "LOGGER_SEVERITY_DEFAULT"
      },{
```

```
        "Id": "TARGET_LOAD",
        "Severity": "LOGGER_SEVERITY_INFO"
      },{
        "Id": "TASK_MANAGER",
        "Severity": "LOGGER_SEVERITY_DEBUG"
      },{
        "Id": "TRANSFORMATION",
        "Severity": "LOGGER_SEVERITY_DEBUG"
      },{
        "Id": "VALIDATOR",
        "Severity": "LOGGER_SEVERITY_DEFAULT"
      }
    ],
    "CloudWatchLogGroup": null,
    "CloudWatchLogStream": null
  },
…
```

Paramètres de tâche de la table de contrôle

Les tables de contrôle fournissent des informations sur une AWS DMS tâche. Elles fournissent également des statistiques utiles que vous pouvez utiliser pour planifier et gérer à la fois la tâche de migration actuelle et les tâches futures. Vous pouvez appliquer ces paramètres de tâche dans un fichier JSON ou en choisissant Paramètres avancés sur la page Créer une tâche de la AWS DMS console. La table Application des exceptions (`dmslogs.awsdms_apply_exceptions`) est toujours créée sur les cibles de base de données. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

AWS DMS crée des tables de contrôle uniquement pendant les tâches Full Load + CDC ou CDC uniquement, et non pendant les tâches Full Load Only.

Pour les tâches de chargement complet + CDC (Migrer les données existantes et répliquer les modifications continues) et les tâches de CDC uniquement (Répliquer les modifications de données uniquement), vous pouvez également créer des tables supplémentaires, notamment les suivantes :

- Statut de réplication (dmslogs.awsdms_status) : cette table fournit des détails sur la tâche en cours. Il s'agit notamment de son statut, de la quantité de mémoire utilisée par la tâche et du nombre de modifications pas encore appliquées à la cible. Ce tableau indique également la position dans la base de données source où la lecture AWS DMS est en cours. Elle indique également si la tâche est en phase de chargement complet ou de capture des données de modification (CDC).

- Tables suspendues (dmslogs.awsdms_suspended_tables) : cette table fournit une liste des tables suspendues, ainsi que la raison pour laquelle elles ont été suspendues.
- Historique de réplication (dmslogs.awsdms_history) : cette table fournit des informations sur l'historique de réplication. Ces informations sont le nombre et le volume d'enregistrements traités pendant la tâche, la latence à la fin d'une tâche de CDC et d'autres statistiques.

La table Application des exceptions (`dmslogs.awsdms_apply_exceptions`) contient les paramètres suivants :

Colonne	Type	Description
TASK_NAME	nvchar	ID de ressource de la AWS DMS tâche. L'ID de ressource se trouve dans l'ARN de la tâche.
TABLE_OWNER	nvchar	Propriétaire de la table.
TABLE_NAME	nvchar	Nom de la table.
ERROR_TIME	timestamp	Heure à laquelle l'exception (erreur) s'est produite.
STATEMENT	nvchar	Déclaration en cours d'exécution lorsque l'erreur s'est produite.
ERROR	nvchar	Nom et description de l'erreur.

La table Statut de réplication (`dmslogs.awsdms_status`) contient le statut actuel de la tâche et de la base de données cible. Ses paramètres sont les suivants :

Colonne	Type	Description
SERVER_NAME	nvchar	Nom de la machine sur laquelle s'exécute la tâche de réplication.

Colonne	Type	Description
TASK_NAME	nvchar	ID de ressource de la AWS DMS tâche. L'ID de ressource se trouve dans l'ARN de la tâche.
TASK_STATUS	varchar	L'une des valeurs suivantes : • FULL LOAD • CHANGE PROCESSING (CDC) • NE FONCTIONNE PAS Le statut de la tâche est défini sur FULL LOAD tant qu'au moins une table est en chargement complet. Une fois que toutes les tables ont été chargées, le statut de la tâche devient CHANGE PROCESSIN G si CDC est activé. La tâche est définie sur NON EXÉCUTABLE avant que vous ne démarriez la tâche ou une fois celle-ci terminée.
STATUS_TIME	timestamp	Horodatage du statut de la tâche.
PENDING_CHANGES	int	Nombre d'enregistrements de modification qui ont été validés dans la base de données source et mis en cache dans la mémoire et le disque de votre instance de réplication.
DISK_SWAP_SIZE	int	Quantité d'espace disque utilisé par des transactions anciennes ou déchargées.
TASK_MEMORY	int	Mémoire actuelle utilisée, en Mo.

Colonne	Type	Description
SOURCE_CURRENT _POSITION	varchar	Position dans la base de données source à partir de laquelle la lecture AWS DMS est en cours.
SOURCE_CURRENT _TIMESTAMP	timestamp	Horodatage de la base de données source en cours AWS DMS de lecture.
SOURCE_TAIL _POSITION	varchar	Position de la transaction de départ la plus ancienne non validée. Cette valeur est la position la plus récente à laquelle vous pouvez revenir sans perdre de modification.
SOURCE_TAIL _TIMESTAMP	timestamp	Horodatage de la transaction de départ la plus ancienne non validée. Cette valeur est l'horodat age le plus récent auquel vous pouvez revenir sans perdre de modification.
SOURCE_TI MESTAMP _APPLIED	timestamp	Horodatage de la dernière validation de transaction. Dans un processus d'application en bloc, cette valeur est l'horodatage de la validation de la dernière transacti on du lot.

La table suspendue (`dmslogs.awsdms_suspended_tables`) contient les paramètres suivants.

Colonne	Type	Description
SERVER_NAME	nvchar	Nom de la machine sur laquelle s'exécute la tâche de réplication.

Colonne	Type	Description
TASK_NAME	nvchar	Le nom de la AWS DMS tâche
TABLE_OWNER	nvchar	Propriétaire de la table.
TABLE_NAME	nvchar	Nom de la table.
SUSPEND_REASON	nvchar	Motif de la suspension.
SUSPEND_TIMESTAMP	timestamp	Heure à laquelle la suspension a eu lieu.

La table Historique de réplication (`dmslogs.awsdms_history`) contient les paramètres suivants :

Colonne	Type	Description
SERVER_NAME	nvchar	Nom de la machine sur laquelle s'exécute la tâche de réplication.
TASK_NAME	nvchar	ID de ressource de la AWS DMS tâche. L'ID de ressource se trouve dans l'ARN de la tâche.
TIMESLOT_TYPE	varchar	L'une des valeurs suivantes : • FULL LOAD • CHANGE PROCESSING (CDC) Si la tâche exécute à la fois les processus de chargement complet et de capture de données modifiées, deux enregistrements d'historique sont consignés dans l'intervalle.
TIMESLOT	timestamp	Horodatage de fin de l'intervalle.

Colonne	Type	Description
TIMESLOT_DURATION	int	Durée de l'intervalle, en minutes.
TIMESLOT_LATENCY	int	Latence cible à la fin de l'intervalle, en secondes. Cette valeur s'applique uniquement aux créneaux horaires CDC.
RECORDS	int	Nombre d'enregistrements traités pendant l'intervalle.
TIMESLOT_VOLUME	int	Volume de données traitées, en Mo.

La table Échec de validation (`awsdms_validation_failures_v1`) contient tous les échecs de validation des données pour une tâche. Pour plus d'informations, consultez Résolution des problèmes liés à la validation des données.

Les paramètres de table de contrôle supplémentaires sont les suivants :

- `HistoryTimeslotInMinutes` : utilisez cette option pour indiquer la longueur de chaque intervalle dans la table Historique de réplication. La valeur par défaut est de 5 minutes.
- `ControlSchema`— Utilisez cette option pour indiquer le nom du schéma de base de données pour les tables de contrôle de la AWS DMS cible. Si vous ne saisissez aucune information pour cette option, les tables sont copiées à l'emplacement par défaut de la base de données comme suit :
  - PostgreSQL, Public
  - Oracle, schéma cible
  - Microsoft SQL Server, dbo dans la base de données cible
  - MySQL, awsdms_control
  - MariaDB, awsdms_control
  - Amazon Redshift, Public
  - DynamoDB, créé en tant que tables individuelles dans la base de données
  - IBM Db2 LUW, awsdms_control

Paramètres de tâche de mémoire tampon de flux

Vous pouvez définir les paramètres de la mémoire tampon du flux à l'aide de AWS CLI, notamment les éléments suivants. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

- `StreamBufferCount` : utilisez cette option pour spécifier le nombre de mémoires tampons de flux de données pour la tâche de migration. Le nombre de mémoires tampon de flux par défaut est 3. L'augmentation de la valeur de ce paramètre pourrait accélérer le processus d'extraction de données. Cependant, cette augmentation des performances dépend fortement de l'environnement de migration, notamment du système source et de la classe d'instance du serveur de réplication. La valeur par défaut est suffisante pour la plupart des cas.
- `StreamBufferSizeInMB` : utilisez cette option pour indiquer la taille maximale de chaque mémoire tampon de flux de données. La taille par défaut est de 8 Mo. Vous devrez peut-être augmenter la valeur pour cette option lorsque vous utiliserez de très grands LOB. Vous devrez peut-être aussi augmenter la valeur si vous recevez un message dans les fichiers-journaux indiquant que la taille de la mémoire tampon de flux est insuffisante. Afin de déterminer la taille de cette option, vous pouvez utiliser l'équation suivante : `[Max LOB size (or LOB chunk size)]*[number of LOB columns]*[number of stream buffers]*[number of tables loading in parallel per task(MaxFullLoadSubTasks)]*3`
- `CtrlStreamBufferSizeInMB` : utilisez cette option pour définir la taille de la mémoire tampon du flux de contrôle. La valeur est exprimée en mégaoctets et peut être comprise entre 1 et 8. La valeur par défaut est 5. Vous devrez peut-être l'augmenter lorsque vous utiliserez un très grand nombre de tables, par exemple des dizaines de milliers de tables.

Paramètres de réglage du traitement des modifications

Les paramètres suivants déterminent le mode de AWS DMS gestion des modifications apportées aux tables cibles lors de la capture des données de modification (CDC). Plusieurs de ces paramètres dépendent de la valeur du paramètre de métadonnées cibles `BatchApplyEnabled`. Pour plus d'informations sur le paramètre `BatchApplyEnabled`, consultez Paramètres de métadonnées des tâches cibles. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

Les paramètres de réglage du traitement des modifications sont les suivants :

Les paramètres ci-dessous s'appliquent uniquement lorsque le paramètre de métadonnées cibles `BatchApplyEnabled` est défini sur `true`.

- `BatchApplyPreserveTransaction` : si cette option est définie sur `true`, l'intégrité transactionnelle est préservée et un lot est assuré de contenir toutes les modifications effectuées dans une transaction à partir de la source. La valeur par défaut est `true`. Ce paramètre s'applique uniquement aux points de terminaison cible Oracle.

  Si elle est définie sur `false`, il peut exister des écarts temporaires dans l'intégrité transactionnelle afin d'améliorer les performances. Il n'existe aucune garantie que toutes les modifications effectuées dans une transaction à partir de la source soient appliquées à la cible en un seul lot.

  Par défaut, AWS DMS les processus changent en mode transactionnel, ce qui préserve l'intégrité des transactions. Si vous pouvez vous permettre des écarts temporaires dans l'intégrité transactionnelle, vous pouvez utiliser l'option d'application par lots optimisée à la place. Cette option regroupe efficacement les transactions et les applique par lots pour plus d'efficacité. L'utilisation de l'option d'application optimisée par lots enfreint presque toujours les contraintes d'intégrité référentielle. Nous vous recommandons donc de désactiver ces contraintes pendant le processus de migration et de les réactiver dans le cadre du processus de basculement.
- `BatchApplyTimeoutMin`— Définit le temps d' AWS DMS attente minimal en secondes entre chaque application de modifications par lots. La valeur par défaut est 1.
- `BatchApplyTimeoutMax`— Définit la durée maximale en secondes qui s' AWS DMS écoule entre chaque application de modifications par lots avant l'expiration du délai. La valeur par défaut est 30.
- `BatchApplyMemoryLimit` : définit la quantité de mémoire maximale (en Mo) à utiliser pour le prétraitement en mode d'application par lots optimisée. La valeur par défaut est 500.
- `BatchSplitSize` : définit le nombre maximal de modifications appliquées dans un même lot. La valeur par défaut est 0, ce qui signifie qu'aucune limite n'est appliquée.

Les paramètres ci-dessous s'appliquent uniquement lorsque le paramètre de métadonnées cibles `BatchApplyEnabled` est défini sur `false`.

- `MinTransactionSize` : définit le nombre minimal de modifications à inclure dans chaque transaction. La valeur par défaut est 1000.
- `CommitTimeout`— Définit le délai maximum en secondes AWS DMS pour collecter les transactions par lots avant de déclarer un délai d'expiration. La valeur par défaut est 1.

Pour la réplication bidirectionnelle, le paramètre suivant s'applique uniquement lorsque le paramètre de métadonnées cibles `BatchApplyEnabled` est défini sur `false`.

- `LoopbackPreventionSettings` : ces paramètres permettent d'empêcher les boucles pour chaque tâche de réplication en cours dans n'importe quelle paire de tâches impliquées dans la réplication bidirectionnelle. La Prévention de boucles empêche l'application de modifications identiques dans les deux sens de la réplication bidirectionnelle, ce qui peut corrompre les données. Pour plus d'informations sur la réplication bidirectionnelle, consultez Effectuer une réplication bidirectionnelle.

AWS DMS tente de conserver les données de transaction en mémoire jusqu'à ce que la transaction soit entièrement validée dans la source, la cible ou les deux. Toutefois, les transactions dont la taille dépasse la mémoire allouée ou qui ne sont pas validées dans le délai imparti sont écrites sur le disque.

Les paramètres suivants s'appliquent au réglage du traitement des modifications quel que soit ce mode de traitement.

- `MemoryLimitTotal` : définit la taille maximale (en Mo) que la totalité des transactions peut occuper en mémoire avant leur écriture sur le disque. La valeur par défaut est 1024.
- `MemoryKeepTime` : définit la durée maximale, en secondes, pendant laquelle chaque transaction peut rester en mémoire avant son écriture sur le disque. La durée est calculée à partir du moment où la capture de la transaction AWS DMS a commencé. La valeur par défaut est 60.
- `StatementCacheSize` : définit le nombre maximal d'instructions préparées à stocker sur le serveur en vue d'une exécution ultérieure lors de l'application des modifications à la cible. La valeur par défaut est 50. La valeur maximale est 200.

Voici un exemple de la façon dont les paramètres de tâche qui gèrent le réglage du traitement des modifications apparaissent dans un fichier JSON de paramètre de tâche :

```
"ChangeProcessingTuning": {
        "BatchApplyPreserveTransaction": true,
        "BatchApplyTimeoutMin": 1,
        "BatchApplyTimeoutMax": 30,
        "BatchApplyMemoryLimit": 500,
        "BatchSplitSize": 0,
        "MinTransactionSize": 1000,
        "CommitTimeout": 1,
        "MemoryLimitTotal": 1024,
        "MemoryKeepTime": 60,
        "StatementCacheSize": 50
```

```
}
```

Pour contrôler la fréquence des écritures sur une cible Amazon S3 lors d'une tâche de réplication de données, vous pouvez configurer les attributs de connexion supplémentaires `cdcMaxBatchInterval` et `cdcMinFileSize`. Cela peut améliorer les performances lors de l'analyse des données sans aucune surcharge supplémentaire. Pour plus d'informations, consultez Paramètres de point de terminaison lors de l'utilisation d'Amazon S3 en tant que cible pour AWS DMS.

Paramètres de la tâche de validation des données

Vous pouvez vérifier que vos données ont été migrées avec précision de la source vers la cible. Si vous activez la validation d'une tâche, AWS DMS commence à comparer les données source et cible immédiatement après le chargement complet d'une table. Pour plus d'informations sur la validation des données d'une tâche, ses exigences, la portée de la prise en charge de sa base de données et les métriques qu'elle fournit, consultez Validation des données AWS DMS. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

Les paramètres de validation des données et leurs valeurs incluent les éléments suivants :

- `EnableValidation` : active la validation des données lorsque la valeur est true. Dans le cas contraire, la validation est désactivée pour la tâche. La valeur par défaut est false.
- `ValidationMode` : contrôle la manière dont DMS validera les données de la table cible par rapport à la table source. AWS DMS fournit ce paramètre pour une extensibilité future. Actuellement, la seule valeur valide par défaut est`ROW_LEVEL`. AWS DMS valide toutes les lignes entre les tables source et cible.
- `FailureMaxCount` : spécifie le nombre maximal d'enregistrements dont la validation peut échouer avant que celle-ci soit suspendue pour la tâche. La valeur par défaut est 10,000. Si vous souhaitez que la validation se poursuive quel que soit le nombre d'enregistrements pour lesquels elle échoue, choisissez une valeur plus élevée que le nombre d'enregistrements dans la source.
- `HandleCollationDiff` : lorsque cette option est définie sur `true`, la validation prend en compte les différences de classement des colonnes dans les points de terminaison PostgreSQL et Microsoft SQL Server lors de l'identification des enregistrements sources et cibles à comparer. Sinon, les différences de classement des colonnes sont ignorées pour la validation. Les classements de colonnes peuvent imposer l'ordre des lignes, ce qui est important pour la validation

des données. Définir `HandleCollationDiff` sur true permet de résoudre automatiquement ces différences de classement et évite les faux positifs dans la validation des données. La valeur par défaut est `false`.

- `RecordFailureDelayInMinutes` : spécifie le délai en minutes avant que les détails d'un échec de validation ne soient fournis.
- `RecordFailureDelayLimitInMinutes` : spécifie le délai avant que les détails d'un échec de validation ne soient fournis. Normalement, AWS DMS utilise la latence de la tâche pour identifier le délai réel pour les modifications à apporter à la cible afin d'éviter les faux positifs. Ce paramètre remplace la valeur de délai réel et vous permet de définir un délai plus important avant la création de rapports sur les métriques de validation. La valeur par défaut est 0.
- `RecordSuspendDelayInMinutes` : spécifie le délai en minutes avant que la validation des tables ne soit suspendue en raison du seuil d'erreur défini dans `FailureMaxCount`.
- `SkipLobColumns`— Lorsque cette option est définie sur`true`, AWS DMS ignore la validation des données pour toutes les colonnes LOB de la partie de la table consacrée à la validation des tâches. La valeur par défaut est `false`.
- `TableFailureMaxCount` : spécifie le nombre maximal de lignes dans une table dont la validation peut échouer avant que la validation soit suspendue pour la table. La valeur par défaut est 1,000.
- `ThreadCount`— Spécifie le nombre de threads d'exécution AWS DMS utilisés lors de la validation. Chaque thread sélectionne les not-yet-validated données de la source et de la cible pour les comparer et les valider. La valeur par défaut est 5. Si vous définissez `ThreadCount` un nombre plus élevé, vous AWS DMS pouvez terminer la validation plus rapidement. Toutefois, AWS DMS exécute ensuite plusieurs requêtes simultanées, consommant plus de ressources sur la source et la cible.
- `ValidationOnly` : lorsque cette option est définie sur `true`, la tâche effectue la validation des données sans aucune migration ou réplication des données. La valeur par défaut est `false`. Vous ne pouvez pas modifier le paramètre `ValidationOnly` une fois la tâche créée.

  Vous devez TargetTablePrepModedéfinir sur `DO_NOTHING` (valeur par défaut pour une tâche de validation uniquement) et définir le type de migration sur l'une des valeurs suivantes :

  - Chargement complet : définissez le type de migration des tâches pour migrer les données existantes dans la AWS DMS console. Ou, dans l' AWS DMS API, définissez le type de migration sur FULL-LOAD.
  - CDC : définissez le Type de migration de la tâche sur Répliquer les modifications de données uniquement dans la console AWS DMS . Ou, dans l' AWS DMS API, définissez le type de migration sur CDC.

Quel que soit le type de migration choisi, les données ne sont pas réellement migrées ou répliquées lors d'une tâche de validation uniquement.

Pour plus d'informations, consultez Tâches de validation uniquement.

> **Important**
>
> Le paramètre `ValidationOnly` est immuable. Il ne peut pas être modifié pour une tâche une fois celle-ci créée.

- `ValidationPartialLobSize` : indique si vous souhaitez effectuer une validation partielle pour les colonnes LOB au lieu de valider toutes les données stockées dans la colonne. Cela peut vous être utile lorsque vous ne migrez qu'une partie des données LOB et non l'ensemble du jeu de données LOB. La valeur est exprimée en Ko. La valeur par défaut est 0, ce qui signifie qu' AWS DMS valide toutes les données de colonne LOB. Par exemple, `"ValidationPartialLobSize": 32` cela AWS DMS ne valide que les 32 premiers Ko des données de colonne à la fois dans la source et dans la cible.
- `PartitionSize` : spécifie la taille du lot d'enregistrements à lire à des fins de comparaison entre la source et la cible. La valeur par défaut est 10 000.
- `ValidationQueryCdcDelaySeconds` : durée pendant laquelle la première requête de validation est retardée à la fois sur la source et sur la cible pour chaque mise à jour de CDC. Cela peut contribuer à réduire la contention des ressources lorsque la latence de migration est élevée. Une tâche de validation uniquement définit automatiquement cette option sur 180 secondes. La valeur par défaut est 0.

Par exemple, le code JSON suivant active la validation des données avec deux fois le nombre de threads par défaut. Elle tient également compte des différences d'ordre d'enregistrement générées par les différences de classement des colonnes dans les points de terminaison PostgreSQL. De plus, elle fournit un délai de rapport de validation pour tenir compte du temps supplémentaire nécessaire pour traiter les défaillances de validation.

```
"ValidationSettings": {
     "EnableValidation": true,
     "ThreadCount": 10,
     "HandleCollationDiff": true,
     "RecordFailureDelayLimitInMinutes": 30
```

```
}
```

> **Note**
>
> Pour un point de terminaison Oracle, AWS DMS utilise DBMS_CRYPTO pour valider les BLOB. Si votre point de terminaison Oracle utilise des objets BLOB, accordez l'autorisation `execute` pour DBMS_CRYPTO au compte d'utilisateur qui accède au point de terminaison Oracle. Pour ce faire, exécutez l'instruction suivante.
>
> ```
> grant execute on sys.dbms_crypto to dms_endpoint_user;
> ```

Paramètres de tâche pour la gestion du DDL de traitement des modifications

Les paramètres suivants déterminent le mode de gestion AWS DMS des modifications du langage de définition des données (DDL) pour les tables cibles lors de la capture des données de modification (CDC). Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

Les paramètres de tâche pour gérer la DDL de traitement des modifications sont les suivants :

- `HandleSourceTableDropped` – Définissez cette option sur `true` pour supprimer la table cible lorsque la table source est supprimée
- `HandleSourceTableTruncated` : définissez cette option sur `true` pour tronquer la table cible lorsque la table source est tronquée.
- `HandleSourceTableAltered` : définissez cette option sur `true` pour modifier la table cible lorsque la table source est modifiée.

Voici un exemple de la façon dont les paramètres de tâche qui gèrent DDL de traitement des modifications apparaissent dans un fichier JSON de paramètre de tâche :

```
            "ChangeProcessingDdlHandlingPolicy": {
               "HandleSourceTableDropped": true,
               "HandleSourceTableTruncated": true,
```

```
            "HandleSourceTableAltered": true
        },
```

> **Note**
>
> Pour en savoir plus sur les instructions DDL prises en charge pour un point de terminaison spécifique, consultez la rubrique décrivant ce point de terminaison.

Paramètres de tâche de substitution de caractères

Vous pouvez spécifier que votre tâche de réplication effectue des substitutions de caractères dans la base de données cible pour toutes les colonnes de la base de données source avec le type de `WSTRING` données AWS DMS `STRING` or. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

Vous pouvez configurer la substitution de caractères pour toute tâche avec des points de terminaison provenant des bases de données source et cible suivantes :

- Bases de données source :
  - Oracle
  - Microsoft SQL Server
  - MySQL
  - PostgreSQL
  - SAP Adaptive Server Enterprise (ASE)
  - IBM Db2 LUW
- Bases de données cible :
  - Oracle
  - Microsoft SQL Server
  - MySQL
  - PostgreSQL
  - SAP Adaptive Server Enterprise (ASE)
  - Amazon Redshift

Vous pouvez spécifier des substitutions de caractères à l'aide du paramètre `CharacterSetSettings` dans vos paramètres de tâche. Ces substitutions de caractères se produisent pour les caractères spécifiés à l'aide de la valeur du point de code Unicode en notation hexadécimale. Vous pouvez implémenter les substitutions en deux phases, dans l'ordre suivant, si les deux sont spécifiées :

1. Remplacement de caractères individuel : AWS DMS permet de remplacer les valeurs des caractères sélectionnés sur la source par des valeurs de remplacement spécifiées des caractères correspondants sur la cible. Utilisez le tableau `CharacterReplacements` dans `CharacterSetSettings` pour sélectionner tous les caractères source ayant les points de code Unicode que vous spécifiez. Utilisez également ce tableau pour spécifier les points de code de remplacement pour les caractères correspondants dans la cible.

   Pour sélectionner tous les caractères de la source qui ont un point de code donné, définissez une instance de `SourceCharacterCodePoint` dans le tableau `CharacterReplacements` sur ce point de code. Spécifiez ensuite le point de code de remplacement pour tous les caractères cibles équivalents en définissant l'instance correspondante de `TargetCharacterCodePoint` dans ce tableau. Pour supprimer les caractères cibles au lieu de les remplacer, définissez les instances appropriées de `TargetCharacterCodePoint` sur zéro (0). Vous pouvez remplacer ou supprimer autant de valeurs différentes de caractères cibles que vous le souhaitez en spécifiant des paires supplémentaires de paramètres `TargetCharacterCodePoint` et `SourceCharacterCodePoint` dans le tableau `CharacterReplacements`. Si vous spécifiez la même valeur pour plusieurs instances de `SourceCharacterCodePoint`, la valeur du dernier paramètre correspondant de `TargetCharacterCodePoint` s'applique à la cible.

   Par exemple, supposons que vous spécifiiez les valeurs suivantes pour `CharacterReplacements`.

```
"CharacterSetSettings": {
    "CharacterReplacements": [ {
        "SourceCharacterCodePoint": 62,
        "TargetCharacterCodePoint": 61
        }, {
        "SourceCharacterCodePoint": 42,
        "TargetCharacterCodePoint": 41
        }
    ]
}
```

Dans cet exemple, AWS DMS remplace tous les caractères ayant la valeur hexadécimale du point de code source 62 sur la cible par des caractères ayant la valeur du point de code 61. AWS DMS Remplace également tous les caractères dont le point de code source est 42 sur la cible par des caractères dont le point de code source est 41. En d'autres termes, AWS DMS remplace toutes les instances de la lettre `'b'` dans la cible par la lettre `'a'`. De même, AWS DMS remplace toutes les instances de la lettre `'B'` sur la cible par la lettre`'A'`.

2. Validation et remplacement du jeu de caractères : une fois le remplacement de chaque caractère effectué, AWS DMS vous pouvez vous assurer que tous les caractères cibles possèdent des points de code Unicode valides dans le jeu de caractères unique que vous spécifiez. Vous utilisez `CharacterSetSupport` dans `CharacterSetSettings` pour configurer cette vérification et cette modification de caractères cibles. Pour spécifier le jeu de caractères de vérification, définissez `CharacterSet` dans `CharacterSetSupport` sur la valeur de chaîne du jeu de caractères lui-même. (Les valeurs possibles pour `CharacterSet` sont indiquées ci-après.) Vous pouvez AWS DMS modifier les caractères cibles non valides de l'une des manières suivantes :

   - Spécifiez un point de code Unicode de remplacement unique pour tous les caractères cibles non valides, quel que soit leur point de code actuel. Pour configurer ce point de code de remplacement, définissez `ReplaceWithCharacterCodePoint` dans `CharacterSetSupport` sur la valeur spécifiée.
   - Configurez la suppression de tous les caractères cibles non valides en définissant `ReplaceWithCharacterCodePoint` sur zéro (0).

   Par exemple, supposons que vous spécifiiez les valeurs suivantes pour `CharacterSetSupport`.

```
"CharacterSetSettings": {
    "CharacterSetSupport": {
        "CharacterSet": "UTF16_PlatformEndian",
        "ReplaceWithCharacterCodePoint": 0
    }
}
```

   Dans cet exemple, AWS DMS supprime tous les caractères trouvés sur la cible qui ne sont pas valides dans le jeu de `"UTF16_PlatformEndian"` caractères. Par conséquent, tous les caractères spécifiés avec la valeur hexadécimale `2FB6` sont supprimés. Cette valeur n'est pas valide, car il s'agit d'un point de code Unicode de 4 octets et les jeux de caractères UTF16 acceptent uniquement les caractères avec des points de code de 2 octets.

> **Note**
>
> La tâche de réplication effectue toutes les substitutions de caractères spécifiées avant de commencer les transformations globales ou au niveau de la table que vous avez spécifiées via le mappage de table. Pour plus d'informations sur le mappage de table, consultez la page Utilisation du mappage de table pour spécifier des paramètres de tâche.
> La substitution de caractères ne prend pas en charge les types de données LOB. Cela s'applique à tous les types de données que DMS considère comme un type de données LOB. Par exemple, le type de données `Extended` dans Oracle est considéré comme des données LOB. Pour plus d'informations sur les types de données source, consultez Types de données sources pour Oracle ci-après.

Les valeurs prises AWS DMS en charge par `CharacterSet` apparaissent dans le tableau suivant.

`UTF-8`	`ibm-860_P100-1995`	`ibm-280_P100-1995`
`UTF-16`	`ibm-861_P100-1995`	`ibm-284_P100-1995`
`UTF-16BE`	`ibm-862_P100-1995`	`ibm-285_P100-1995`
`UTF-16LE`	`ibm-863_P100-1995`	`ibm-290_P100-1995`
`UTF-32`	`ibm-864_X110-1999`	`ibm-297_P100-1995`
`UTF-32BE`	`ibm-865_P100-1995`	`ibm-420_X120-1999`
`UTF-32LE`	`ibm-866_P100-1995`	`ibm-424_P100-1995`
`UTF16_PlatformEndian`	`ibm-867_P100-1998`	`ibm-500_P100-1995`
`UTF16_OppositeEndian`	`ibm-868_P100-1995`	`ibm-803_P100-1999`
`UTF32_PlatformEndian`	`ibm-869_P100-1995`	`ibm-838_P100-1995`
`UTF32_OppositeEndian`	`ibm-878_P100-1996`	`ibm-870_P100-1995`
`UTF-16BE,version=1`	`ibm-901_P100-1999`	`ibm-871_P100-1995`
`UTF-16LE,version=1`	`ibm-902_P100-1999`	`ibm-875_P100-1995`

UTF-16,version=1	ibm-922_P100-1999	ibm-918_P100-1995
UTF-16,version=2	ibm-1168_P100-2002	ibm-930_P120-1999
UTF-7	ibm-4909_P100-1999	ibm-933_P110-1995
IMAP-mailbox-name	ibm-5346_P100-1998	ibm-935_P110-1999
SCSU	ibm-5347_P100-1998	ibm-937_P110-1999
BOCU-1	ibm-5348_P100-1997	ibm-939_P120-1999
CESU-8	ibm-5349_P100-1998	ibm-1025_P100-1995
ISO-8859-1	ibm-5350_P100-1998	ibm-1026_P100-1995
US-ASCII	ibm-9447_P100-2002	ibm-1047_P100-1995
gb18030	ibm-9448_X100-2005	ibm-1097_P100-1995
ibm-912_P100-1995	ibm-9449_P100-2002	ibm-1112_P100-1995
ibm-913_P100-2000	ibm-5354_P100-1998	ibm-1122_P100-1999
ibm-914_P100-1995	ibm-1250_P100-1995	ibm-1123_P100-1995
ibm-915_P100-1995	ibm-1251_P100-1995	ibm-1130_P100-1997
ibm-1089_P100-1995	ibm-1252_P100-2000	ibm-1132_P100-1998
ibm-9005_X110-2007	ibm-1253_P100-1995	ibm-1137_P100-1999
ibm-813_P100-1995	ibm-1254_P100-1995	ibm-4517_P100-2005
ibm-5012_P100-1999	ibm-1255_P100-1995	ibm-1140_P100-1997
ibm-916_P100-1995	ibm-5351_P100-1998	ibm-1141_P100-1997
ibm-920_P100-1995	ibm-1256_P110-1997	ibm-1142_P100-1997
iso-8859_10-1998	ibm-5352_P100-1998	ibm-1143_P100-1997

iso-8859_11-2001	ibm-1257_P100-1995	ibm-1144_P100-1997
ibm-921_P100-1995	ibm-5353_P100-1998	ibm-1145_P100-1997
iso-8859_14-1998	ibm-1258_P100-1997	ibm-1146_P100-1997
ibm-923_P100-1998	macos-0_2-10.2	ibm-1147_P100-1997
ibm-942_P12A-1999	macos-6_2-10.4	ibm-1148_P100-1997
ibm-943_P15A-2003	macos-7_3-10.2	ibm-1149_P100-1997
ibm-943_P130-1999	macos-29-10.2	ibm-1153_P100-1999
ibm-33722_P12A_P12A-2009_U2	macos-35-10.2	ibm-1154_P100-1999
ibm-33722_P120-1999	ibm-1051_P100-1995	ibm-1155_P100-1999
ibm-954_P101-2007	ibm-1276_P100-1995	ibm-1156_P100-1999
euc-jp-2007	ibm-1006_P100-1995	ibm-1157_P100-1999
ibm-1373_P100-2002	ibm-1098_P100-1995	ibm-1158_P100-1999
windows-950-2000	ibm-1124_P100-1996	ibm-1160_P100-1999
ibm-950_P110-1999	ibm-1125_P100-1997	ibm-1164_P100-1999
ibm-1375_P100-2008	ibm-1129_P100-1997	ibm-1364_P110-2007
ibm-5471_P100-2006	ibm-1131_P100-1997	ibm-1371_P100-1999
ibm-1386_P100-2001	ibm-1133_P100-1997	ibm-1388_P103-2001
windows-936-2000	ISO_2022,locale=ja,version=0	ibm-1390_P110-2003
ibm-1383_P110-1999	ISO_2022,locale=ja,version=1	ibm-1399_P110-2003

ibm-5478_P100-1995	ISO_2022,locale=ja,version=2	ibm-5123_P100-1999
euc-tw-2014	ISO_2022,locale=ja,version=3	ibm-8482_P100-1999
ibm-964_P110-1999	ISO_2022,locale=ja,version=4	ibm-16684_P110-2003
ibm-949_P110-1999	ISO_2022,locale=ko,version=0	ibm-4899_P100-1998
ibm-949_P11A-1999	ISO_2022,locale=ko,version=1	ibm-4971_P100-1999
ibm-970_P110_P110-2006_U2	ISO_2022,locale=zh,version=0	ibm-9067_X100-2005
ibm-971_P100-1995	ISO_2022,locale=zh,version=1	ibm-12712_P100-1998
ibm-1363_P11B-1998	ISO_2022,locale=zh,version=2	ibm-16804_X110-1999
ibm-1363_P110-1997	HZ	ibm-37_P100-1995,swaplfnl
windows-949-2000	x11-compound-text	ibm-1047_P100-1995,swaplfnl
windows-874-2000	ISCII,version=0	ibm-1140_P100-1997,swaplfnl
ibm-874_P100-1995	ISCII,version=1	ibm-1141_P100-1997,swaplfnl
ibm-1162_P100-1999	ISCII,version=2	ibm-1142_P100-1997,swaplfnl

ibm-437_P100-1995	ISCII,version=3	ibm-1143_P100-1997,swaplfnl
ibm-720_P100-1997	ISCII,version=4	ibm-1144_P100-1997,swaplfnl
ibm-737_P100-1997	ISCII,version=5	ibm-1145_P100-1997,swaplfnl
ibm-775_P100-1996	ISCII,version=6	ibm-1146_P100-1997,swaplfnl
ibm-850_P100-1995	ISCII,version=7	ibm-1147_P100-1997,swaplfnl
ibm-851_P100-1995	ISCII,version=8	ibm-1148_P100-1997,swaplfnl
ibm-852_P100-1995	LMBCS-1	ibm-1149_P100-1997,swaplfnl
ibm-855_P100-1995	ibm-37_P100-1995	ibm-1153_P100-1999,swaplfnl
ibm-856_P100-1995	ibm-273_P100-1995	ibm-12712_P100-1998,swaplfnl
ibm-857_P100-1995	ibm-277_P100-1995	ibm-16804_X110-1999,swaplfnl
ibm-858_P100-1997	ibm-278_P100-1995	ebcdic-xml-us

Paramètres de tâche de l'image antérieure

Lorsque vous écrivez des mises à jour de CDC sur une cible de diffusion de données comme Kinesis ou Apache Kafka, vous pouvez afficher les valeurs d'origine d'une ligne de base de données source avant de les modifier par une mise à jour. Pour ce faire, AWS DMS remplit une image antérieure des événements de mise à jour en fonction des données fournies par le moteur de base de données

source. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

Pour ce faire, utilisez le paramètre `BeforeImageSettings` qui ajoute un nouvel attribut JSON à chaque opération de mise à jour avec des valeurs collectées dans le système de base de données source.

Assurez-vous d'appliquer `BeforeImageSettings` uniquement aux tâches de chargement complet + CDC ou aux tâches de CDC uniquement. Les tâches de chargement complet + CDC migrent les données existantes et répliquent les modifications continues. Les tâches de CDC uniquement répliquent les modifications de données uniquement.

N'appliquez pas les `BeforeImageSettings` aux tâches à pleine charge uniquement.

Les options admises pour `BeforeImageSettings` sont les suivantes :

- `EnableBeforeImage` : lorsque cette option est définie sur `true`, active la génération d'image antérieure. L'argument par défaut est `false`.
- `FieldName` : affecte un nom au nouvel attribut JSON. Quand `EnableBeforeImage` est `true`, `FieldName` est obligatoire et ne peut pas être vide.
- `ColumnFilter` : spécifie une colonne à ajouter en utilisant la génération d'image antérieure. Pour ajouter uniquement des colonnes faisant partie des clés primaires de la table, utilisez la valeur par défaut, `pk-only`. Pour ajouter une colonne ayant une valeur d'image antérieure, utilisez `all`. Notez que l'image antérieure ne prend pas en charge les types de données LOB (objets binaires de grande taille) tels que CLOB et BLOB.

L'exemple suivant illustre l'utilisation de `BeforeImageSettings`.

```
"BeforeImageSettings": {
    "EnableBeforeImage": true,
    "FieldName": "before-image",
    "ColumnFilter": "pk-only"
  }
```

Pour en savoir plus sur les paramètres d'image antérieure pour Kinesis, y compris sur les paramètres supplémentaires de mappage de table, consultez Utilisation d'une image antérieure pour afficher les valeurs d'origine des lignes CDC pour un flux de données Kinesis en tant que cible.

Pour plus d'informations sur les paramètres d'image antérieure pour Kafka, y compris les paramètres supplémentaires de mappage de table, consultez Utilisation d'une image antérieure pour afficher les valeurs originales des lignes de la CDC pour Apache Kafka comme cible.

Paramètres de traitement des erreurs pour une tâche

Vous pouvez définir le comportement de gestion des erreurs de votre tâche de réplication en utilisant les paramètres suivants. Pour en savoir plus sur l'utilisation d'un fichier de configuration de tâche pour définir les paramètres d'une tâche, consultez Exemple de paramètres de tâche.

- `DataErrorPolicy`— Détermine l'action AWS entreprise par DMS en cas d'erreur liée au traitement des données au niveau de l'enregistrement. Certains exemples d'erreurs liées au traitement de données comprennent les erreurs de conversion, les erreurs dans la transformation et les données incorrectes. L'argument par défaut est `LOG_ERROR`.
  - `IGNORE_RECORD` : la tâche continue et les données de cet enregistrement sont ignorées. Le compteur d'erreurs pour la propriété `DataErrorEscalationCount` est incrémenté. Par conséquent, si vous définissez une limite pour les erreurs pour une table, cette erreur est prise en compte dans cette limite.
  - `LOG_ERROR` : la tâche continue et l'erreur est écrite dans le journal de la tâche.
  - `SUSPEND_TABLE` : la tâche continue, mais les données de la table avec l'enregistrement d'erreur basculent à l'état d'erreur et les données ne sont pas répliquées.
  - `STOP_TASK` : la tâche s'arrête et une intervention manuelle est requise.
- `DataTruncationErrorPolicy` : détermine l'action exécutée par AWS DMS lorsque les données sont tronquées. L'argument par défaut est `LOG_ERROR`.
  - `IGNORE_RECORD` : la tâche continue et les données de cet enregistrement sont ignorées. Le compteur d'erreurs pour la propriété `DataErrorEscalationCount` est incrémenté. Par conséquent, si vous définissez une limite pour les erreurs pour une table, cette erreur est prise en compte dans cette limite.
  - `LOG_ERROR` : la tâche continue et l'erreur est écrite dans le journal de la tâche.
  - `SUSPEND_TABLE` : la tâche continue, mais les données de la table avec l'enregistrement d'erreur basculent à l'état d'erreur et les données ne sont pas répliquées.
  - `STOP_TASK` : la tâche s'arrête et une intervention manuelle est requise.
- `DataErrorEscalationPolicy` : détermine l'action exécutée par AWS DMS lorsque le nombre maximal d'erreurs (défini dans le paramètre `DataErrorEscalationCount`) est atteint. L'argument par défaut est `SUSPEND_TABLE`.

  - SUSPEND_TABLE : la tâche continue, mais les données de la table avec l'enregistrement d'erreur basculent à l'état d'erreur et les données ne sont pas répliquées.
  - STOP_TASK : la tâche s'arrête et une intervention manuelle est requise.
- DataErrorEscalationCount : définit le nombre maximal d'erreurs qui peuvent avoir lieu sur les données pour un enregistrement spécifique. Lorsque ce nombre est atteint, les données de la table qui contient l'enregistrement d'erreur sont traitées conformément à la stratégie définie dans DataErrorEscalationPolicy. La valeur par défaut est 0.
- EventErrorPolicy— Détermine l'action entreprise par AWS DMS lorsqu'une erreur se produit lors de l'envoi d'un événement lié à une tâche. Les valeurs possibles sont les suivantes :
  - IGNORE : la tâche continue et toutes les données associées à cet événement sont ignorées.
  - STOP_TASK : la tâche s'arrête et une intervention manuelle est requise.
- TableErrorPolicy : détermine l'action exécutée par AWS DMS lorsqu'une erreur se produit pendant le traitement des données ou des métadonnées d'une table spécifique. Cette erreur s'applique uniquement aux données générales de la table. Il ne s'agit pas d'une erreur liée à un enregistrement spécifique. L'argument par défaut est SUSPEND_TABLE.
  - SUSPEND_TABLE : la tâche continue, mais les données de la table avec l'enregistrement d'erreur basculent à l'état d'erreur et les données ne sont pas répliquées.
  - STOP_TASK : la tâche s'arrête et une intervention manuelle est requise.
- TableErrorEscalationPolicy : détermine l'action exécutée par AWS DMS lorsque le nombre maximal d'erreurs (défini à l'aide du paramètre TableErrorEscalationCount) est atteint. Le paramètre par défaut et le seul paramètre utilisateur est STOP_TASK, avec lequel la tâche est arrêtée et une intervention manuelle est requise.
- TableErrorEscalationCount : nombre maximal d'erreurs qui peuvent avoir lieu sur les données ou les métadonnées générales pour une table spécifique. Lorsque ce nombre est atteint, les données de la table sont traitées conformément à la stratégie définie dans TableErrorEscalationPolicy. La valeur par défaut est 0.
- RecoverableErrorCount : nombre maximal de tentatives effectuées pour redémarrer une tâche lorsqu'une erreur d'environnement se produit. Une fois que le système a tenté de redémarrer la tâche le nombre de fois indiqué, la tâche est arrêtée et une intervention manuelle est requise. La valeur par défaut est -1, ce qui indique AWS DMS de tenter de redémarrer la tâche indéfiniment. Définissez cette valeur sur 0 pour ne jamais essayer de redémarrer une tâche. En cas d'erreur fatale, AWS DMS arrête la tentative de redémarrage de la tâche après six tentatives.
- RecoverableErrorInterval— Le nombre de secondes pendant lesquelles le AWS DMS attend entre deux tentatives de redémarrage d'une tâche. La valeur par défaut est 5.

- `RecoverableErrorThrottling` : lorsque cette option est activée, l'intervalle entre les tentatives de redémarrage d'une tâche est augmenté en série en fonction de la valeur de `RecoverableErrorInterval`. Par exemple, si `RecoverableErrorInterval` est défini sur 5 secondes, la prochaine tentative aura lieu après 10 secondes, puis 20, puis 40 secondes et ainsi de suite. L'argument par défaut est `true`.
- `RecoverableErrorThrottlingMax`— Le nombre maximal de secondes pendant lesquelles le AWS DMS attend entre deux tentatives de redémarrage d'une tâche s'il `RecoverableErrorThrottling` est activé. La valeur par défaut est 1800.
- `RecoverableErrorStopRetryAfterThrottlingMax`— Lorsque ce paramètre est défini sur`true`, arrête le redémarrage de la tâche une fois que le nombre maximum de secondes d' AWS DMS attente entre les tentatives de restauration est atteint, par. `RecoverableErrorThrottlingMax`
- `ApplyErrorDeletePolicy` : détermine l'action exécutée par AWS DMS quand un conflit lié à une opération DELETE a lieu. L'argument par défaut est `IGNORE_RECORD`. Les valeurs possibles sont les suivantes :
  - `IGNORE_RECORD` : la tâche continue et les données de cet enregistrement sont ignorées. Le compteur d'erreurs pour la propriété `ApplyErrorEscalationCount` est incrémenté. Par conséquent, si vous définissez une limite pour les erreurs pour une table, cette erreur est prise en compte dans cette limite.
  - `LOG_ERROR` : la tâche continue et l'erreur est écrite dans le journal de la tâche.
  - `SUSPEND_TABLE` : la tâche continue, mais les données de la table avec l'enregistrement d'erreur basculent à l'état d'erreur et les données ne sont pas répliquées.
  - `STOP_TASK` : la tâche s'arrête et une intervention manuelle est requise.
- `ApplyErrorInsertPolicy` : détermine l'action exécutée par AWS DMS quand un conflit lié à une opération INSERT a lieu. L'argument par défaut est `LOG_ERROR`. Les valeurs possibles sont les suivantes :
  - `IGNORE_RECORD` : la tâche continue et les données de cet enregistrement sont ignorées. Le compteur d'erreurs pour la propriété `ApplyErrorEscalationCount` est incrémenté. Par conséquent, si vous définissez une limite pour les erreurs pour une table, cette erreur est prise en compte dans cette limite.
  - `LOG_ERROR` : la tâche continue et l'erreur est écrite dans le journal de la tâche.
  - `SUSPEND_TABLE` : la tâche continue, mais les données de la table avec l'enregistrement d'erreur basculent à l'état d'erreur et les données ne sont pas répliquées.
  - `STOP_TASK` : la tâche s'arrête et une intervention manuelle est requise.

  - `INSERT_RECORD` : s'il existe un enregistrement cible avec la même clé primaire que l'enregistrement source inséré, l'enregistrement cible est mis à jour.
- `ApplyErrorUpdatePolicy` : détermine l'action exécutée par AWS DMS quand un conflit de données manquantes lié à une opération UPDATE a lieu. L'argument par défaut est `LOG_ERROR`. Les valeurs possibles sont les suivantes :
  - `IGNORE_RECORD` : la tâche continue et les données de cet enregistrement sont ignorées. Le compteur d'erreurs pour la propriété `ApplyErrorEscalationCount` est incrémenté. Par conséquent, si vous définissez une limite pour les erreurs pour une table, cette erreur est prise en compte dans cette limite.
  - `LOG_ERROR` : la tâche continue et l'erreur est écrite dans le journal de la tâche.
  - `SUSPEND_TABLE` : la tâche continue, mais les données de la table avec l'enregistrement d'erreur basculent à l'état d'erreur et les données ne sont pas répliquées.
  - `STOP_TASK` : la tâche s'arrête et une intervention manuelle est requise.
  - `UPDATE_RECORD`— Si l'enregistrement cible est manquant, l'enregistrement cible manquant est inséré dans la table cible. AWS DMS désactive complètement le support des colonnes LOB pour la tâche. La sélection de cette option nécessite qu'une journalisation supplémentaire complète soit activée pour toutes les colonnes de la table source quand Oracle est la base de données source.
- `ApplyErrorEscalationPolicy`— Détermine l'action AWS entreprise par DMS lorsque le nombre maximum d'erreurs (défini à l'aide du `ApplyErrorEscalationCount` paramètre) est atteint. La valeur par défaut est LOG_ERROR :
  - `LOG_ERROR` : la tâche continue et l'erreur est écrite dans le journal de la tâche.
  - `SUSPEND_TABLE` : la tâche continue, mais les données de la table avec l'enregistrement d'erreur basculent à l'état d'erreur et les données ne sont pas répliquées.
  - `STOP_TASK` : la tâche s'arrête et une intervention manuelle est requise.
- `ApplyErrorEscalationCount` : cette option définit le nombre maximal de conflits APPLY qui peuvent se produire pour une table spécifique pendant une opération de traitement des modifications. Lorsque ce nombre est atteint, les données de la table sont traitées conformément à la stratégie définie dans le paramètre `ApplyErrorEscalationPolicy`. La valeur par défaut est 0.
- `ApplyErrorFailOnTruncationDdl` : définissez cette option sur `true` pour faire échouer la tâche lorsqu'une troncation est effectuée sur l'une des tables suivies pendant la CDC. L'argument par défaut est `false`.

Cette approche ne fonctionne pas avec PostgreSQL version 11.x ou antérieures, ou tout autre point de terminaison source qui ne réplique pas la troncation de table DDL.

- `FailOnNoTablesCaptured` : définissez cette option sur `true` pour faire échouer une tâche lorsque les mappages de table définis pour une tâche ne trouvent pas de tables au lancement de la tâche. L'argument par défaut est `false`.
- `FailOnTransactionConsistencyBreached` : cette option s'applique aux tâches qui utilisent Oracle en tant que source avec la CDC. La valeur par défaut est false. Définissez le paramètre sur `true` pour qu'une tâche échoue lorsqu'une transaction est ouverte plus longtemps que le délai d'expiration spécifié et puisse être supprimée.

  Lorsqu'une tâche CDC démarre avec Oracle, attendez AWS DMS pendant une durée limitée que la transaction ouverte la plus ancienne soit clôturée avant de démarrer CDC. Si la transaction ouverte la plus ancienne ne se ferme pas avant que le délai ne soit atteint, le CDC AWS DMS démarre dans la plupart des cas, en ignorant cette transaction. Si cette option est définie sur `true`, la tâche échoue.
- `FullLoadIgnoreConflicts`— Définissez cette option `true` pour AWS DMS ignorer les erreurs « zéro ligne affectée » et « doublons » lors de l'application d'événements mis en cache. Si ce paramètre est défini AWS DMS sur`false`, signale toutes les erreurs au lieu de les ignorer. L'argument par défaut est `true`.

Notez que les erreurs de chargement de table dans Redshift en tant que cible sont signalées dans `STL_LOAD_ERRORS`. Pour plus d'informations, consultez STL_LOAD_ERRORS dans le Guide du développeur de base de données Amazon Redshift.

Paramètres de tâche d'enregistrement

Vous pouvez enregistrer les paramètres de tâche en tant que fichier JSON, au cas où vous souhaiteriez réutiliser les paramètres pour une autre tâche. Vous trouverez les paramètres de tâche à copier dans un fichier JSON dans la section Détails de présentation d'une tâche.

> **Note**
>
> Lorsque vous réutilisez les paramètres de tâche pour d'autres tâches, supprimez tous les attributs `CloudWatchLogGroup` et `CloudWatchLogStream`. Sinon, l'erreur suivante s'affiche : MESSAGE D'ERREUR SYSTÈME : Paramètres des tâches CloudWatchLogGroup ou CloudWatchLogStream impossible à définir lors de la création.

Par exemple, le fichier JSON suivant contient les paramètres enregistrés pour une tâche.

```
{
    "TargetMetadata": {
        "TargetSchema": "",
        "SupportLobs": true,
        "FullLobMode": false,
        "LobChunkSize": 0,
        "LimitedSizeLobMode": true,
        "LobMaxSize": 32,
        "InlineLobMaxSize": 0,
        "LoadMaxFileSize": 0,
        "ParallelLoadThreads": 0,
        "ParallelLoadBufferSize": 0,
        "BatchApplyEnabled": false,
        "TaskRecoveryTableEnabled": false,
        "ParallelLoadQueuesPerThread": 0,
        "ParallelApplyThreads": 0,
        "ParallelApplyBufferSize": 0,
        "ParallelApplyQueuesPerThread": 0
    },
    "FullLoadSettings": {
        "TargetTablePrepMode": "DO_NOTHING",
        "CreatePkAfterFullLoad": false,
        "StopTaskCachedChangesApplied": false,
        "StopTaskCachedChangesNotApplied": false,
        "MaxFullLoadSubTasks": 8,
        "TransactionConsistencyTimeout": 600,
        "CommitRate": 10000
    },
    "Logging": {
        "EnableLogging": true,
        "LogComponents": [
            {
                "Id": "TRANSFORMATION",
                "Severity": "LOGGER_SEVERITY_DEFAULT"
            },
            {
                "Id": "SOURCE_UNLOAD",
                "Severity": "LOGGER_SEVERITY_DEFAULT"
            },
            {
                "Id": "IO",
```

```
            "Severity": "LOGGER_SEVERITY_DEFAULT"
        },
        {
            "Id": "TARGET_LOAD",
            "Severity": "LOGGER_SEVERITY_DEFAULT"
        },
        {
            "Id": "PERFORMANCE",
            "Severity": "LOGGER_SEVERITY_DEFAULT"
        },
        {
            "Id": "SOURCE_CAPTURE",
            "Severity": "LOGGER_SEVERITY_DEFAULT"
        },
        {
            "Id": "SORTER",
            "Severity": "LOGGER_SEVERITY_DEFAULT"
        },
        {
            "Id": "REST_SERVER",
            "Severity": "LOGGER_SEVERITY_DEFAULT"
        },
        {
            "Id": "VALIDATOR_EXT",
            "Severity": "LOGGER_SEVERITY_DEFAULT"
        },
        {
            "Id": "TARGET_APPLY",
            "Severity": "LOGGER_SEVERITY_DEFAULT"
        },
        {
            "Id": "TASK_MANAGER",
            "Severity": "LOGGER_SEVERITY_DEFAULT"
        },
        {
            "Id": "TABLES_MANAGER",
            "Severity": "LOGGER_SEVERITY_DEFAULT"
        },
        {
            "Id": "METADATA_MANAGER",
            "Severity": "LOGGER_SEVERITY_DEFAULT"
        },
        {
            "Id": "FILE_FACTORY",
```

```
                "Severity": "LOGGER_SEVERITY_DEFAULT"
            },
            {
                "Id": "COMMON",
                "Severity": "LOGGER_SEVERITY_DEFAULT"
            },
            {
                "Id": "ADDONS",
                "Severity": "LOGGER_SEVERITY_DEFAULT"
            },
            {
                "Id": "DATA_STRUCTURE",
                "Severity": "LOGGER_SEVERITY_DEFAULT"
            },
            {
                "Id": "COMMUNICATION",
                "Severity": "LOGGER_SEVERITY_DEFAULT"
            },
            {
                "Id": "FILE_TRANSFER",
                "Severity": "LOGGER_SEVERITY_DEFAULT"
            }
        ]
    },
    "ControlTablesSettings": {
        "ControlSchema": "",
        "HistoryTimeslotInMinutes": 5,
        "HistoryTableEnabled": false,
        "SuspendedTablesTableEnabled": false,
        "StatusTableEnabled": false,
        "FullLoadExceptionTableEnabled": false
    },
    "StreamBufferSettings": {
        "StreamBufferCount": 3,
        "StreamBufferSizeInMB": 8,
        "CtrlStreamBufferSizeInMB": 5
    },
    "ChangeProcessingDdlHandlingPolicy": {
        "HandleSourceTableDropped": true,
        "HandleSourceTableTruncated": true,
        "HandleSourceTableAltered": true
    },
    "ErrorBehavior": {
        "DataErrorPolicy": "LOG_ERROR",
```

```
        "DataTruncationErrorPolicy": "LOG_ERROR",
        "DataErrorEscalationPolicy": "SUSPEND_TABLE",
        "DataErrorEscalationCount": 0,
        "TableErrorPolicy": "SUSPEND_TABLE",
        "TableErrorEscalationPolicy": "STOP_TASK",
        "TableErrorEscalationCount": 0,
        "RecoverableErrorCount": -1,
        "RecoverableErrorInterval": 5,
        "RecoverableErrorThrottling": true,
        "RecoverableErrorThrottlingMax": 1800,
        "RecoverableErrorStopRetryAfterThrottlingMax": true,
        "ApplyErrorDeletePolicy": "IGNORE_RECORD",
        "ApplyErrorInsertPolicy": "LOG_ERROR",
        "ApplyErrorUpdatePolicy": "LOG_ERROR",
        "ApplyErrorEscalationPolicy": "LOG_ERROR",
        "ApplyErrorEscalationCount": 0,
        "ApplyErrorFailOnTruncationDdl": false,
        "FullLoadIgnoreConflicts": true,
        "FailOnTransactionConsistencyBreached": false,
        "FailOnNoTablesCaptured": true
    },
    "ChangeProcessingTuning": {
        "BatchApplyPreserveTransaction": true,
        "BatchApplyTimeoutMin": 1,
        "BatchApplyTimeoutMax": 30,
        "BatchApplyMemoryLimit": 500,
        "BatchSplitSize": 0,
        "MinTransactionSize": 1000,
        "CommitTimeout": 1,
        "MemoryLimitTotal": 1024,
        "MemoryKeepTime": 60,
        "StatementCacheSize": 50
    },
    "PostProcessingRules": null,
    "CharacterSetSettings": null,
    "LoopbackPreventionSettings": null,
    "BeforeImageSettings": null,
    "FailTaskWhenCleanTaskResourceFailed": false
}
```

Configuration du support LOB pour les bases de données sources dans une tâche AWS DMS

Les grands objets binaires (LOB) peuvent parfois être difficiles à migrer d'un système à l'autre. AWS DMS propose un certain nombre d'options pour faciliter l'ajustement de colonnes LOB. Pour savoir quels types de données sont considérés comme des LOB et à quel moment AWS DMS, consultez la AWS DMS documentation.

Lorsque vous migrez des données d'une base de données vers une autre, vous pouvez saisir l'occasion pour repenser la manière dont vos LOB sont stockés, en particulier pour les migrations hétérogènes. Si vous souhaitez le faire, vous n'avez pas besoin de migrer les données LOB.

Si vous choisissez d'inclure les objets LOB, vous pouvez choisir ensuite les autres paramètres LOB :

- Le mode LOB détermine la façon dont les objets LOB sont gérés :
  - Mode LOB complet : en mode LOB complet, tous les LOB AWS DMS migrent de la source vers la cible, quelle que soit leur taille. Dans cette configuration, il n'y AWS DMS a aucune information sur la taille maximale des LOB à attendre. Par conséquent, les LOB sont migrés un par un, morceau par morceau. Le mode LOB complet peut être très lent.
  - Mode LOB limité : en mode LOB limité, vous définissez une taille maximale d'objet LOB devant être acceptée par DMS. Cela permet à DMS de pré-allouer de la mémoire et de charger les données LOB en bloc. Les LOB qui dépassent la taille de LOB maximale sont tronqués et un avertissement est publié dans le fichier journal. En mode LOB limité, vous pouvez bénéficier de gains de performance significatifs par rapport au mode LOB complet. Nous vous recommandons d'utiliser le mode LOB limité dans la mesure du possible. La valeur maximale recommandée est de 102 400 Ko (100 Mo).

    > **Note**
    >
    > Si vous utilisez l'option Taille de LOB maximale (Ko) avec une valeur supérieure à 63 Ko, cela aura un impact sur les performances d'une charge complète configurée pour s'exécuter en mode LOB limité. Lors d'un chargement complet, DMS alloue de la mémoire en multipliant la valeur de l'option Taille de LOB maximale (Ko) par le taux de validation ; le produit est alors multiplié par le nombre de colonnes LOB. Lorsque DMS ne peut pas préallouer cette mémoire, DMS commence à consommer de la mémoire SWAP, ce qui a un impact sur les performances d'un chargement complet. Par conséquent, si vous rencontrez des problèmes de performances lorsque vous utilisez le mode LOB limité, envisagez de réduire le taux de validation jusqu'à atteindre un niveau

de performance acceptable. Vous pouvez également envisager d'utiliser le mode LOB en ligne pour les points de terminaison pris en charge une fois que vous aurez identifié la distribution LOB de la table.
Pour valider la taille de LOB limitée, vous devez définir `ValidationPartialLobSize` sur la même valeur que `LobMaxSize` (Ko).

- Mode LOB en ligne : en mode LOB en ligne, vous définissez la taille de LOB maximale que DMS transfère en ligne. Les objets LOB inférieurs à la taille spécifiée sont transférés en ligne. Les objets LOB supérieurs à la taille spécifiée sont répliqués en mode LOB complet. Vous pouvez sélectionner cette option pour répliquer à la fois des LOB de petite et de grande taille lorsque la plupart des objets LOB sont de petite taille. DMS ne prend pas en charge le mode LOB en ligne pour les points de terminaison qui ne prennent pas en charge le mode LOB complet, tels que S3 et Redshift.

> **Note**
>
> Avec Oracle, les LOB sont traités comme des types de données VARCHAR si possible. Cette approche signifie qu'elles sont AWS DMS extraites de la base de données en masse, ce qui est nettement plus rapide que les autres méthodes. La taille maximale d'un VARCHAR dans Oracle est de 32 Ko. Par conséquent, une taille LOB limitée inférieure à 32 Ko est optimale si Oracle est la base de données source.

- Lorsqu'une tâche est configurée pour s'exécuter en Max LOB size (K) (Taille maximale du LOB (K)), l'option détermine la taille de LOB maximale acceptée par AWS DMS . Les grands objets (LOB) qui dépassent cette valeur sont tronqués à cette valeur.
- Lorsqu'une tâche est configurée pour utiliser le mode LOB complet, elle AWS DMS récupère les LOB en plusieurs parties. L'option Taille de bloc du LOB (ko) détermine la taille de chaque élément. Lorsque vous définissez cette option, portez une attention particulière à la taille de paquet maximale autorisée par votre configuration réseau. Si la taille de bloc LOB dépasse la taille maximale autorisée pour les paquets, vous pourriez voir des erreurs de déconnexion. La valeur recommandée pour `LobChunkSize` est de 64 kilo-octets. L'augmentation de la valeur de `LobChunkSize` au-delà de 64 kilo-octets peut entraîner l'échec des tâches.
- Lorsqu'une tâche est configurée pour s'exécuter en mode LOB en ligne, le paramètre `InlineLobMaxSize` détermine les objets LOB que DMS transfère en ligne.

Pour plus d'informations sur les paramètres de tâche pour spécifier ces options, consultez Paramètres de métadonnées des tâches cibles

Création de plusieurs tâches

Dans certains scénarios de migration, vous pouvez avoir à créer plusieurs tâches de migration. Les tâches fonctionnent indépendamment les unes des autres et peuvent s'exécuter simultanément. Chaque tâche possède ses propres chargement initial, CDC et procédure de lecture des journaux. Les tables reliées par le biais du langage de manipulation de données (DML) doivent faire partie de la même tâche.

Voici quelques raisons de créer plusieurs tâches pour une migration :

- Les tables cibles pour les tâches résident dans différentes bases de données, par exemple lorsque vous distribuez ou divisez un système en plusieurs systèmes.
- Vous souhaitez diviser la migration d'une grande table en plusieurs tâches grâce au filtrage.

> **Note**
>
> Étant donné que chaque tâche possède son propre processus de capture des modifications et de lecture de journaux, les modifications ne sont pas coordonnées entre les tâches. Par conséquent, lorsque vous utilisez plusieurs tâches pour effectuer une migration, assurez-vous que chaque transaction source individuelle est entièrement contenue dans une tâche unique. Vous pouvez utiliser plusieurs tâches pour effectuer une migration si aucune transaction individuelle n'est répartie entre différentes tâches.

Création de tâches pour la réplication continue à l'aide d'AWS DMS

Vous pouvez créer une tâche AWS DMS qui capture les modifications en continu à partir du magasin de données source. Pour ce faire, vous pouvez capturer pendant que vous migrez vos données. Vous pouvez également créer une tâche qui capture les modifications continues une fois que vous avez terminé la migration initiale (chargement complet) vers un magasin de données cible pris en charge. Ce processus est appelé la réplication continue ou capture des données modifiées (CDC). AWS DMS utilise ce processus lors de la réplication des modifications continues à partir d'un magasin de données source. Lors de ce processus, les modifications apportées aux journaux de base de données sont collectées à l'aide de l'API native du moteur de base de données.

> **Note**
>
> Vous pouvez migrer des vues à l'aide de tâches de chargement complet uniquement. Si votre tâche est une tâche CDC uniquement ou une tâche de chargement complet qui démarre la capture des données modifiées (CDC) une fois terminée, la migration inclut uniquement les tables de la source. Si vous utilisez une tâche de chargement complet uniquement, vous pouvez migrer des vues ou une combinaison de tables et de vues. Pour de plus amples informations, veuillez consulter Spécification des règles de sélection de table et de transformation à l'aide de JSON.

Chaque moteur source possède une configuration requise spécifique pour communiquer ce flux de modifications à un compte d'utilisateur donné. La plupart des moteurs ont besoin d'une configuration supplémentaire pour que le processus de capture puisse exploiter les données modifiées de manière significative, sans perte de données. Par exemple, Oracle nécessite l'ajout d'une journalisation supplémentaire et MySQL nécessite une journalisation binaire au niveau des lignes.

Pour lire les modifications continues à partir de la base de données source, AWS DMS utilise des actions d'API propres au moteur pour lire les modifications à partir des journaux de transactions du moteur source. Voici quelques exemples montrant la façon dont AWS DMS exécute cette opération :

- Pour Oracle, AWS DMS utilise l'API Oracle LogMiner ou Binary Reader (API bfile) pour lire les modifications en continu. AWS DMS lit les modifications en continu dans les journaux redo en ligne ou archivés en fonction du SCN (numéro de modification système).
- Pour Microsoft SQL Server, AWS DMS utilise MS-Replication ou MS-CDC pour écrire des informations dans le journal des transactions SQL Server. Il utilise ensuite la fonction `fn_dblog()` ou `fn_dump_dblog()` dans SQL Server pour lire les modifications dans le journal des transactions en fonction du LSN (numéro de séquence de journal).
- Pour MySQL, AWS DMS lit les modifications à partir des journaux binaires basés sur les lignes et migre ces modifications vers la cible.
- Pour PostgreSQL, AWS DMS configure des emplacements de réplication logique et utilise le plug-in test_decoding pour lire les modifications à partir de la source et les migrer vers la cible.
- Pour Amazon RDS en tant que source, nous vous recommandons de vérifier que les sauvegardes sont activées pour configurer la CDC. Nous vous recommandons également de vous assurer que la base de données source est configurée pour conserver les journaux de modification pendant une durée suffisante (24 heures suffisent généralement). Pour obtenir les paramètres spécifiques à chaque point de terminaison, consultez :

- Amazon RDS for Oracle : Configuration d'une source Oracle AWS gérée pour AWS DMS.
- Amazon RDS for MySQL et Aurora MySQL : Utilisation d'une base de données compatible MySQL gérée par AWS en tant que source pour AWS DMS.
- Amazon RDS for SQL Server : Configuration de la réplication continue sur une instance de base de données SQL Server cloud.
- Amazon RDS for PostgreSQL et Aurora PostgreSQL : PostgreSQL conserve automatiquement les journaux WAL requis.

Il existe deux types de tâches de réplication continue :

- Chargement complet + CDC : la tâche migre les données existantes, puis met à jour la base de données cible en fonction des modifications apportées à la base de données source.
- CDC uniquement : la tâche migre les modifications continues une fois que les données sont sur la base de données cible.

Exécution de la réplication à partir d'un point de départ CDC

Vous pouvez démarrer une tâche de réplication continue AWS DMS (tâche de capture des données modifiées (CDC) uniquement) à partir de plusieurs points, Tel est le cas des éléments suivants :

- À partir d'une heure de début de CDC personnalisée : vous pouvez utiliser AWS Management Console ou AWS CLI pour fournir à AWS DMS un horodatage auquel vous voulez démarrer la réplication. AWS DMS démarre alors une tâche de réplication continue à partir de cette heure de début de CDC personnalisée. AWS DMS convertit l'horodatage donné (au format UTC) en point de départ natif, par exemple un LSN pour SQL Server ou un SCN pour Oracle. AWS DMS utilise des méthodes spécifiques à chaque moteur pour déterminer où démarrer la tâche de migration en fonction du flux de modifications du moteur source.

> **Note**
>
> Ce n'est qu'en définissant l'attribut de connexion `StartFromContext` sur l'horodatage requis que DB2, en tant que source, propose une heure de début de CDC personnalisée. PostgreSQL comme source ne prend pas en charge une heure de début CDC personnalisée. Cela est dû au fait que le moteur de base de données PostgreSQL n'a

aucun moyen de faire correspondre un horodatage à un LSN ou un SCN, à la différence d'Oracle et SQL Server.

- À partir d'un point de départ natif de CDC : vous pouvez également démarrer à partir d'un point natif dans le journal des transactions du moteur source. Dans certains cas, cette approche peut être préférable, car un horodatage peut indiquer plusieurs points natifs dans le journal des transactions. AWS DMS prend en charge cette fonctionnalité pour les points de terminaison source suivants :
  - SQL Server
  - PostgreSQL
  - Oracle
  - MySQL
  - MariaDB

Lorsque la tâche est créée, AWS DMS marque le point de départ de la CDC et il ne peut pas être modifié. Pour utiliser un point de départ de CDC différent, créez une nouvelle tâche.

Définition d'un point de départ natif CDC

Un point de départ natif CDC est un point dans le journal du moteur de base de données qui définit un moment auquel vous pouvez commencer la capture des données modifiées (CDC). À titre d'exemple, supposons qu'un vidage des données en bloc a déjà été appliqué à la cible. Vous pouvez rechercher le point de départ natif de la tâche de réplication continue uniquement. Pour éviter toute incohérence dans les données, choisissez avec soin le point de départ de la tâche de réplication uniquement. DMS capture les transactions qui ont débuté après le point de départ de CDC choisi.

Voici des exemples montrant comment trouver le point de départ natif CDC à partir de moteurs sources pris en charge :

SQL Server

Dans SQL Server, un LSN (numéro de séquence de journal) comporte trois parties :

- Numéro de séquence VLF (fichier journal virtuel)
- Décalage de départ d'un bloc de journaux
- Numéro de l'emplacement

Voici un exemple de LSN : `00000014:00000061:0001`

Pour obtenir le point de départ d'une tâche de migration SQL Server en fonction de vos paramètres de sauvegarde du journal des transactions, utilisez la fonction `fn_dblog()` ou `fn_dump_dblog()` dans SQL Server.

Pour utiliser le point de départ natif de CDC avec SQL Server, créez une publication sur n'importe quelle table participant à la réplication continue. AWS DMS crée la publication automatiquement lorsque vous utilisez la CDC sans utiliser de point de départ natif de CDC.

PostgreSQL

Vous utilisez un point de contrôle de récupération CDC pour votre base de données source PostgreSQL. Cette valeur de point de contrôle est générée en différents points pendant l'exécution d'une tâche de réplication continue pour votre base de données source (la tâche parent). Pour plus d'informations sur les points de contrôle en général, consultez Utilisation d'un point de contrôle comme point de départ CDC.

Pour identifier le point de contrôle à utiliser comme point de départ natif, utilisez la vue `pg_replication_slots` de la base de données ou les détails de présentation de la tâche parent dans AWS Management Console.

Pour trouver les détails de la présentation de votre tâche parent sur la console

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS DMS à l'adresse https://console.aws.amazon.com/dms/v2/.

   Si vous êtes connecté en tant qu'utilisateur IAM, assurez-vous d'avoir les autorisations appropriées pour accéder à AWS DMS. Pour plus d'informations sur les autorisations requises, consultez Autorisations IAM nécessaires pour utiliser AWS DMS.
2. Dans le volet de navigation de la console, choisissez, Tâches de migration de base de données.
3. Choisissez votre tâche parent dans la liste de la page Tâches de migration de base de données. Vous ouvrez ainsi la page de la tâche parent et affichez les détails de la présentation.
4. Trouvez la valeur du point de contrôle sous Capture de données modifiées (CDC), Position de départ de la capture de données modifiées (CDC) et Point de contrôle de récupération de la capture des données modifiées (CDC).

   La valeur qui apparaît est similaire celle ci-dessous.

```
checkpoint:V1#1#000004AF/B00000D0#0#0#*#0#0
```

Ici, le composant `4AF/B00000D0` est ce dont vous avez besoin pour spécifier ce point de départ CDC natif. Définissez le paramètre `CdcStartPosition` de l'API DMS sur cette valeur lorsque vous créez la tâche CDC pour commencer la réplication à ce point de départ pour votre source PostgreSQL. Pour de plus amples informations sur l'utilisation de l'AWS CLI pour créer cette tâche CDC, veuillez consulter Activation de la CDC avec une instance de base de données PostgreSQL gérée par AWS avec AWS DMS.

Oracle

Un SCN (numéro de modification système) est un horodatage interne, logique, utilisé par les bases de données Oracle. Les SCN classent les événements qui se produisent au sein de la base de données, ce qui est nécessaire pour répondre aux propriétés ACID d'une transaction. Les bases de données Oracle utilisent des SCN pour marquer l'emplacement où toutes les modifications ont été écrites sur le disque afin qu'une action de récupération n'applique pas les modifications déjà écrites. Oracle utilise également les SCN pour marquer le point où il n'existe pas de redo pour un ensemble de données, afin que la récupération puisse s'arrêter.

Pour obtenir le SCN actuel dans une base de données Oracle, exécutez la commande suivante.

```
SELECT CURRENT_SCN FROM V$DATABASE
```

Si vous utilisez le SCN ou l'horodatage pour démarrer une tâche de CDC, les résultats des transactions ouvertes sont manquants et vous ne parvenez pas à les migrer. Les transactions ouvertes sont des transactions qui ont été démarrées avant la position de départ de la tâche et validées après la position de départ de la tâche. Vous pouvez identifier le SCN et l'horodatage pour démarrer une tâche de CDC à un point qui inclut toutes les transactions ouvertes. Pour plus d'informations, consultez Transactions dans la documentation d'Oracle en ligne. Avec la version 3.5.1 et les versions ultérieures, AWS DMS prend en charge les transactions ouvertes pour une tâche de CDC uniquement en utilisant le paramètre de point de terminaison `openTransactionWindow` si vous utilisez le SCN ou l'horodatage pour démarrer la tâche.

Lorsque vous utilisez le paramètre `openTransactionWindow`, vous devez fournir la fenêtre (en minutes) dans laquelle gérer les transactions ouvertes. AWS DMS décale la position de capture et trouve la nouvelle position pour démarrer la capture des données. AWS DMS utilise la nouvelle

position de départ pour analyser toutes les transactions ouvertes à partir des journaux redo Oracle requis ou des journaux redo archivés.

MySQL

Avant la version MySQL 5.6.3, le LSN pour MySQL était un entier non signé de 4 octets. Dans la version MySQL 5.6.3, lorsque la limite de taille du fichier journal redo est passée de 4 Go à 512 Go, le LSN est devenu un entier non signé à 8 octets. L'augmentation indique qu'il était nécessaire d'augmenter le nombre d'octets pour stocker des informations supplémentaires sur la taille. Les applications reposant sur MySQL 5.6.3 ou versions ultérieures et qui utilisent les LSN doivent utiliser des variables de 64 bits plutôt que 32 bits pour stocker et comparer les valeurs LSN. Pour plus d'informations sur les LSN MySQL, consultez la documentation MySQL.

Pour obtenir le LSN actuel dans une base de données MySQL, exécutez la commande suivante.

```
mysql> show master status;
```

La requête renvoie le nom d'un fichier binlog, la position, ainsi que plusieurs autres valeurs. Le point de départ natif CDC est une combinaison du nom du fichier binlog et de la position, par exemple `mysql-bin-changelog.000024:373`. Dans cet exemple, `mysql-bin-changelog.000024` est le nom du fichier binlog et `373`, la position où AWS DMS doit démarrer la capture des modifications.

Utilisation d'un point de contrôle comme point de départ CDC

Une tâche de réplication continue migre les modifications, et AWS DMS met de temps en temps en cache des informations spécifiques de AWS DMS. Le point de contrôle créé par AWS DMS contient des informations qui permettent au moteur de réplication de connaître le point de récupération pour le flux des modifications. Vous pouvez utiliser le point de contrôle pour revenir dans la chronologie des modifications et récupérer une tâche de migration ayant échoué. Vous pouvez également utiliser un point de contrôle pour démarrer une autre tâche de réplication continue pour une autre cible à tout moment.

Vous pouvez obtenir les informations de point de contrôle de l'une des trois manières suivantes :

- Exécutez l'opération d'API `DescribeReplicationTasks` et consultez les résultats. Vous pouvez filtrer les informations par tâche et rechercher le point de contrôle. Vous pouvez récupérer le dernier point de contrôle lorsque la tâche est arrêtée ou en échec. Ces informations sont perdues si la tâche est supprimée.

- Affichez la table de métadonnées nommée `awsdms_txn_state` sur l'instance cible. Vous pouvez interroger la table pour obtenir les informations de point de contrôle. Pour créer la table de métadonnées, définissez le paramètre `TaskRecoveryTableEnabled` sur `Yes` lorsque vous créez une tâche. Ce paramètre indique à AWS DMS d'écrire en continu des informations de point de contrôle dans la table de métadonnées cible. Ces informations sont perdues si une tâche est supprimée.

  Voici un exemple de point de contrôle dans la table de métadonnées :
  `checkpoint:V1#34#00000132/0F000E48#0#0#*#0#121`
- Dans le volet de navigation, choisissez Tâches de migration de base de données, puis choisissez votre tâche parent dans la liste qui apparaît sur la page Tâches de migration de base de données. La page de votre tâche parent s'ouvre avec les détails de présentation. Trouvez la valeur du point de contrôle sous Capture de données modifiées (CDC), Position de départ de la capture de données modifiées (CDC) et Point de contrôle de récupération de la capture des données modifiées (CDC). La valeur de point de contrôle qui apparaît est similaire à ce qui suit :

  `checkpoint:V1#1#000004AF/B00000D0#0#0#*#0#0`

Arrêt d'une tâche à un point de validation ou de serveur

Grâce à l'introduction des points de départ CDC natifs, AWS DMS peut également arrêter une tâche aux points suivants :

- Heure de validation sur la source
- Heure du serveur sur l'instance de réplication

Vous pouvez modifier une tâche et définir une heure au format UTC pour arrêter au moment souhaité. La tâche s'arrête automatiquement en fonction de l'heure de validation ou du serveur que vous définissez. Par ailleurs, si vous souhaitez arrêter la tâche de migration à une heure précise au moment de la création de la tâche, vous pouvez définir une heure d'arrêt lorsque vous créez la tâche.

> **Note**
>
> L'initialisation de toutes les ressources peut prendre jusqu'à 40 minutes la première fois que vous démarrez une nouvelle réplication AWS DMS sans serveur. Notez que l'option `server_time` n'est applicable qu'une fois l'initialisation des ressources terminée.

Effectuer une réplication bidirectionnelle

Vous pouvez utiliser des tâches AWS DMS pour effectuer une réplication bidirectionnelle entre deux systèmes. Dans la réplication bidirectionnelle, vous répliquez des données de la même table (ou d'un ensemble de tables) entre deux systèmes dans les deux sens.

Par exemple, vous pouvez copier une table EMPLOYE de la base de données A vers la base de données B et répliquer les modifications apportées à la table de la base de données A vers la base de données B. Vous pouvez également répliquer les modifications apportées à la table EMPLOYE de la base de données B vers A. Ainsi, vous effectuez une réplication bidirectionnelle.

> **Note**
>
> La réplication bidirectionnelle AWS DMS n'est pas conçue comme une solution multi-maître complète incluant un nœud principal, la résolution des conflits, etc.

Utilisez la réplication bidirectionnelle pour des cas dans lesquels les données sur différents nœuds sont séparées au niveau opérationnel. En d'autres termes, supposons qu'un élément de données soit modifié par une application fonctionnant sur le nœud A, et que ce dernier effectue une réplication bidirectionnelle avec le nœud B. Cet élément de données sur le nœud A n'est jamais modifié par une application fonctionnant sur le nœud B.

AWS DMS prend en charge la réplication bidirectionnelle sur ces moteurs de base de données :

- Oracle
- SQL Server
- MySQL
- PostgreSQL
- Amazon Aurora MySQL-Compatible Edition
- Aurora PostgreSQL-Compatible Edition

Création de tâches de réplication bidirectionnelle

Pour activer la réplication bidirectionnelle AWS DMS, configurez les points de terminaison source et cible pour les deux bases de données (A et B). Par exemple, configurez un point de terminaison

source pour la base de données A, un point de terminaison source pour la base de données B, un point de terminaison cible pour la base de données A et un point de terminaison cible pour la base de données B.

Ensuite, créez deux tâches : une tâche pour la source A pour déplacer les données vers la cible B, et une autre tâche pour la source B pour déplacer les données vers la cible A. Assurez-vous également que chaque tâche est configurée pour empêcher les boucles. Vous pourrez ainsi empêcher l'application de modifications identiques aux cibles des deux tâches, ce qui corrompt les données d'au moins une d'entre elles. Pour de plus amples informations, veuillez consulter Empêcher les boucles.

Pour l'approche la plus simple, commencez par des jeux de données identiques sur les bases de données A et B. Ensuite, créez deux tâches CDC uniquement, une tâche pour répliquer les données de A vers B et une autre tâche pour répliquer les données de B vers A.

Pour instancier un nouveau jeu de données (base de données) sur le nœud B à partir du nœud A à l'aide de AWS DMS, procédez comme suit :

1. Utilisez une charge complète et une tâche CDC pour déplacer les données de la base de données A vers B. Assurez-vous qu'aucune application ne modifie les données de la base de données B pendant ce temps.
2. Lorsque la charge complète est terminée et avant que les applications ne soient autorisées à modifier les données de la base de données B, notez l'heure ou la position de départ de la CDC de la base de données B. Pour obtenir des instructions, veuillez consulter Exécution de la réplication à partir d'un point de départ CDC.
3. Créez une tâche CDC uniquement qui déplace les données de la base de données B vers A à l'aide de cette heure de début ou de cette position de départ CDC.

> **Note**
>
> Une seule tâche dans une paire bidirectionnelle peut être totalement chargée et CDC.

Empêcher les boucles

Pour afficher la prévention des boucles, supposons que dans une tâche T1 AWS DMS lit les journaux des modifications à partir de la base de données source A et applique les modifications à la base de données cible B.

Ensuite, une deuxième tâche, T2, lit les journaux des modifications à partir de la base de données source B et les applique à la base de données cible A. Avant que T2 ne fasse cela, DMS doit s'assurer que les modifications apportées à la base de données cible B à partir de la base de données source A ne sont pas apportées à la base de données source A. En d'autres termes, DMS doit s'assurer que ces modifications ne sont pas reprises (en boucle) dans la base de données cible A. Dans le cas contraire, les données de la base de données A peuvent être corrompues.

Pour éviter les boucles de modifications, ajoutez les paramètres de tâche suivants à chaque tâche de réplication bidirectionnelle. Ainsi, vous vous assurez que la corruption des données pour cause de boucles ne se produit dans aucun sens.

```
{
. . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": Boolean,
    "SourceSchema": String,
    "TargetSchema": String
  },

. . .
}
```

Les paramètres de tâche `LoopbackPreventionSettings` déterminent si une transaction est nouvelle ou reprise de la tâche de réplication opposée. Lorsqu'une transaction est appliquée par AWS DMS à une base de données cible, elle met à jour une table DMS (`awsdms_loopback_prevention`) avec une indication de la modification. Avant d'appliquer chaque transaction à une cible, DMS ignore toute transaction qui inclut une référence à cette table `awsdms_loopback_prevention`. Par conséquent, il n'applique pas la modification.

Inclure ces paramètres de tâche avec chaque tâche de réplication dans une paire bidirectionnelle. Ces paramètres permettent la prévention des boucles. Ils spécifient également le schéma de chaque base de données source et cible dans la tâche qui inclut la table `awsdms_loopback_prevention` pour chaque point de terminaison.

Pour permettre à chaque tâche d'identifier une reprise et de le supprimer, définissez `EnableLoopbackPrevention` sur `true`. Pour spécifier un schéma dans la source qui inclut `awsdms_loopback_prevention`, définissez `SourceSchema` sur le nom de ce schéma dans la base de données source. Pour spécifier un schéma sur la cible qui inclut la même table, définissez `TargetSchema` sur le nom de ce schéma dans la base de données cible.

Dans l'exemple suivant, les paramètres `TargetSchema` et `SourceSchema` pour une tâche de réplication T1 et sa tâche de réplication opposée T2 sont spécifiés avec des paramètres opposés.

Les paramètres de la tâche T1 sont les suivants.

```
{
. . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": true,
    "SourceSchema": "LOOP-DATA",
    "TargetSchema": "loop-data"
  },

. . .
}
```

Les paramètres de la tâche opposée T2 sont les suivants.

```
{
. . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": true,
    "SourceSchema": "loop-data",
    "TargetSchema": "LOOP-DATA"
  },

. . .
}
```

> **Note**
>
> Lorsque vous utilisez l'AWS CLI, utilisez uniquement les commandes `modify-replication-task` ou `create-replication-task` pour configurer `LoopbackPreventionSettings` dans vos tâches de réplication bidirectionnelle.

Limites de la réplication bidirectionnelle

La réplication bidirectionnelle pour AWS DMS présente les limites suivantes :

- La prévention des boucles suit uniquement les instructions en langage de manipulation de données (DML). AWS DMS ne prend pas en charge la prévention des boucles de langage de définition de données (DDL). Pour ce faire, configurez l'une des tâches dans une paire bidirectionnelle pour filtrer les instructions DDL.
- Les tâches qui utilisent la prévention de boucles ne prennent pas en charge la validation des modifications dans des lots. Pour configurer une tâche avec la prévention de boucles, assurez-vous de définir `BatchApplyEnabled` sur `false`.
- La réplication bidirectionnelle DMS n'inclut pas la détection ou la résolution des conflits. Pour détecter des incohérences de données, utilisez la validation des données sur les deux tâches.

Modification d'une tâche

Vous pouvez modifier une tâche si vous avez besoin changer des paramètres de tâche, un mappage de table ou d'autres paramètres. Vous pouvez également activer et exécuter des évaluations de prémigration avant d'exécuter la tâche modifiée. Vous pouvez modifier une tâche dans la console en sélectionnant la tâche et en choisissant Modifier. Vous pouvez également utiliser la commande CLI ou l'opération d'API ModifyReplicationTask.

Quelques limitations s'appliquent à la modification d'une tâche, Tel est le cas des éléments suivants :

- Vous ne pouvez pas modifier le point de terminaison source ou cible d'une tâche.
- Vous ne pouvez pas modifier le type de migration d'une tâche.
- Les tâches qui ont été exécutées doivent avoir le statut Arrêté ou Échec pour être modifiées.

Déplacement d'une tâche

Vous pouvez déplacer une tâche vers une autre instance de réplication lorsque l'une des situations suivantes s'applique à votre cas d'utilisation.

- Vous utilisez actuellement une instance d'un certain type et vous souhaitez passer à un autre type d'instance.
- Votre instance actuelle est surchargée par de nombreuses tâches de réplication et vous souhaitez diviser la charge entre plusieurs instances.
- Le stockage de votre instance est plein et vous souhaitez déplacer les tâches de cette instance vers une instance plus puissante au lieu de mettre à l'échelle le stockage ou le calcul.

- Vous souhaitez utiliser une fonctionnalité nouvellement publiée d'AWS DMS, mais vous ne voulez pas créer de nouvelle tâche ni redémarrer la migration. Vous préférez plutôt mettre en service une instance de réplication avec une nouvelle version d'AWS DMS prenant en charge cette fonctionnalité, et déplacer la tâche existante vers cette instance.

Vous pouvez déplacer une tâche dans la console en sélectionnant la tâche, puis en choisissant Déplacer. Vous pouvez également utiliser la commande CLI ou l'opération d'API `MoveReplicationTask` pour déplacer la tâche. Vous pouvez déplacer une tâche dont le point de terminaison cible est un moteur de base de données quelconque.

Veillez à ce que l'instance de réplication cible dispose de suffisamment d'espace de stockage pour prendre en charge la tâche déplacée. Sinon, mettez à l'échelle le stockage pour libérer de l'espace pour votre instance de réplication cible avant de déplacer la tâche.

Veillez également à ce que votre instance de réplication cible soit créée avec une version du moteur AWS DMS égale ou ultérieure à celle de l'instance de réplication actuelle.

> **Note**
>
> - Vous ne pouvez pas déplacer une tâche vers l'instance de réplication où elle réside actuellement.
> - Vous ne pouvez pas modifier les paramètres d'une tâche lorsqu'elle est en cours de déplacement.
> - Une tâche que vous avez exécutée doit avoir le statut Arrêté, Échec ou Échec du déplacement pour que vous puissiez la déplacer.

Deux statuts de tâche sont liés au déplacement d'une tâche DMS : Déplacement et Échec du déplacement. Pour plus d'informations sur le statut d'une tâche, consultez État de la tâche.

Après avoir déplacé une tâche, vous pouvez activer et exécuter des évaluations de prémigration pour rechercher des problèmes de blocage avant d'exécuter la tâche déplacée.

Rechargement de tables pendant une tâche

Pendant l'exécution d'une tâche, vous pouvez recharger une base de données cible à l'aide de données provenant de la source. Vous pouvez souhaiter recharger une table si, pendant la tâche,

une erreur se produit ou si des données sont modifiées à cause d'opérations de partition (par exemple, lors de l'utilisation d'Oracle). Vous pouvez recharger jusqu'à 10 tables à partir d'une tâche.

Le rechargement des tables n'arrête pas la tâche.

Pour recharger une table, les conditions suivantes doivent être appliquées :

- La tâche doit être en cours d'exécution.
- La méthode de migration de la tâche doit être Full Load ou Full Load with CDC.
- Les doublons de tables ne sont pas autorisés.
- AWS DMS conserve la définition de table lue précédemment et ne la recrée pas au cours de l'opération de rechargement. Toutes les instructions DDL telles que ALTER TABLE ADD COLUMN, DROP COLUMN apportées à la table avant le rechargement de la table peuvent entraîner l'échec de l'opération de rechargement.

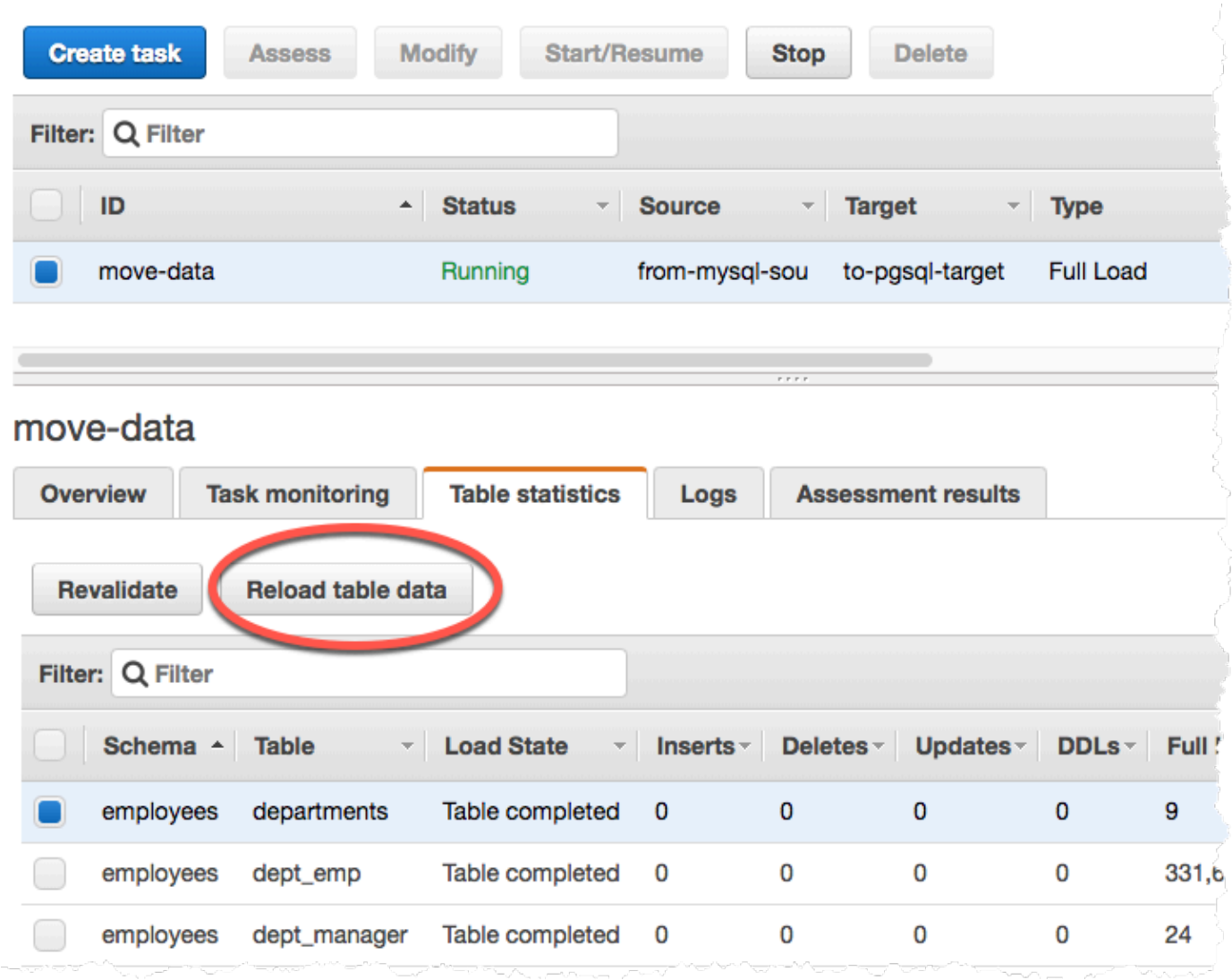
> **Note**
>
> DMS applique le paramètre `TargetTablePrepMode` avant de recharger la table. Si vous définissez `TargetTablePrepMode` sur `DO_NOTHING`, vous devez d'abord tronquer manuellement la table.

AWS Management Console

Pour recharger une table à l'aide de la console AWS DMS

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS DMS à l'adresse https://console.aws.amazon.com/dms/v2/.

   Si vous êtes connecté en tant qu'utilisateur IAM, assurez-vous d'avoir les autorisations appropriées pour accéder à AWS DMS. Pour plus d'informations sur les autorisations requises, consultez Autorisations IAM nécessaires pour utiliser AWS DMS.
2. Choisissez Tâches dans le volet de navigation.
3. Choisissez la tâche en cours d'exécution contenant la table que vous souhaitez recharger.
4. Choisissez l'onglet Statistiques de table.



5. Choisissez la table que vous souhaitez recharger. Si la tâche n'est plus en cours d'exécution, vous ne pouvez pas recharger la table.
6. Choisissez Recharger les données de la table.

Lorsqu'AWS DMS se prépare à recharger une table, la console modifie l'état de la table, qui devient Table is being reloaded (La table est en cours de rechargement).

Utilisation du mappage de table pour spécifier des paramètres de tâche

Le mappage de table utilise différents types de règles pour spécifier la source de données, le schéma source, les données et toutes les transformations qui doivent se produire pendant la tâche. Vous

pouvez utiliser le mappage de table pour spécifier des tables individuelles d'une base de données à migrer et le schéma à utiliser pour la migration.

Lorsque vous utilisez le mappage de table, vous pouvez utiliser des filtres pour spécifier les données à répliquer à partir des colonnes de table. De plus, vous pouvez utiliser des transformations pour modifier les schémas, tables ou vues sélectionnés avant qu'ils soient écrits dans la base de données cible.

Rubriques

- Spécification des règles de sélection de table et de transformation à partir de la console
- Spécification des règles de sélection de table et de transformation à l'aide de JSON
- Règles et actions de sélection
- Caractères génériques dans le mappage de table
- Règles et actions de transformation
- Utilisation d'expressions de règle de transformation pour définir le contenu d'une colonne
- Règles des paramètres de table et de collection et opérations

> **Note**
>
> Lorsque vous utilisez le mappage de table pour un point de terminaison source MongoDB, vous pouvez utiliser des filtres pour spécifier les données que vous souhaitez répliquer, et spécifier un nom de base de données à la place de `schema_name`. Sinon, vous pouvez utiliser la valeur `"%"` par défaut.

Spécification des règles de sélection de table et de transformation à partir de la console

Vous pouvez utiliser le AWS Management Console pour effectuer un mappage de table, notamment pour spécifier la sélection de tables et les transformations. Dans la console, utilisez la section Where pour spécifier le schéma, la table et l'action (include ou exclude). Utilisez la section Filter pour spécifier le nom de la colonne dans une table et les conditions que vous souhaitez appliquer à une tâche de réplication. Ensemble, ces deux actions créent une règle de sélection.

Vous pouvez inclure des transformations dans un mappage de table une fois que vous avez spécifié au moins une règle de sélection. Vous pouvez utiliser les transformations pour renommer un schéma

ou une table, ajouter un préfixe ou un suffixe à un schéma ou une table ou supprimer une colonne d'une table.

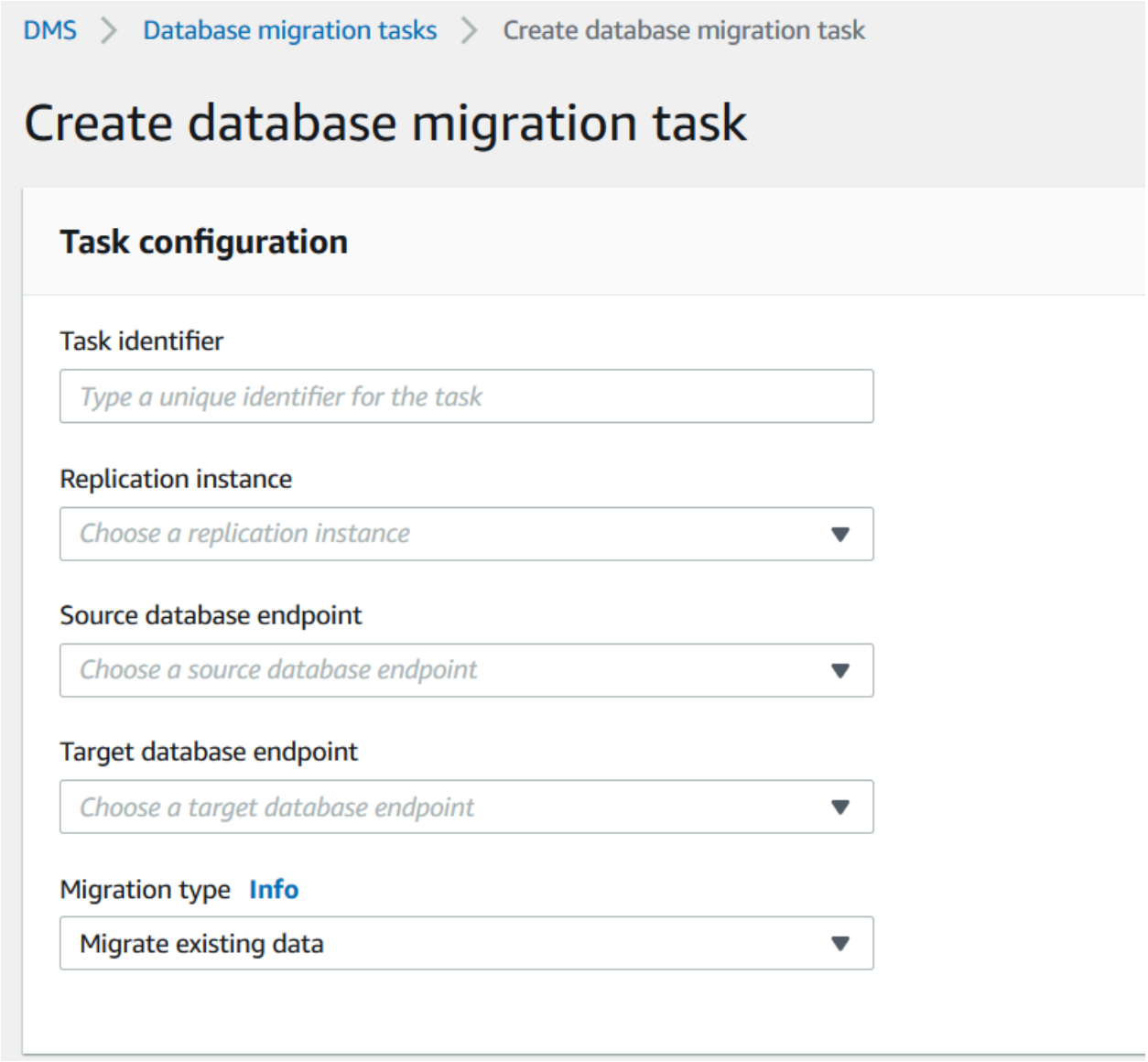
> **Note**
>
> AWS DMS ne prend pas en charge plus d'une règle de transformation par niveau de schéma, de table ou de colonne.

La procédure suivante montre comment configurer des règles de sélection, basées sur une table appelée **`Customers`** dans un schéma appelé **`EntertainmentAgencySample`**.

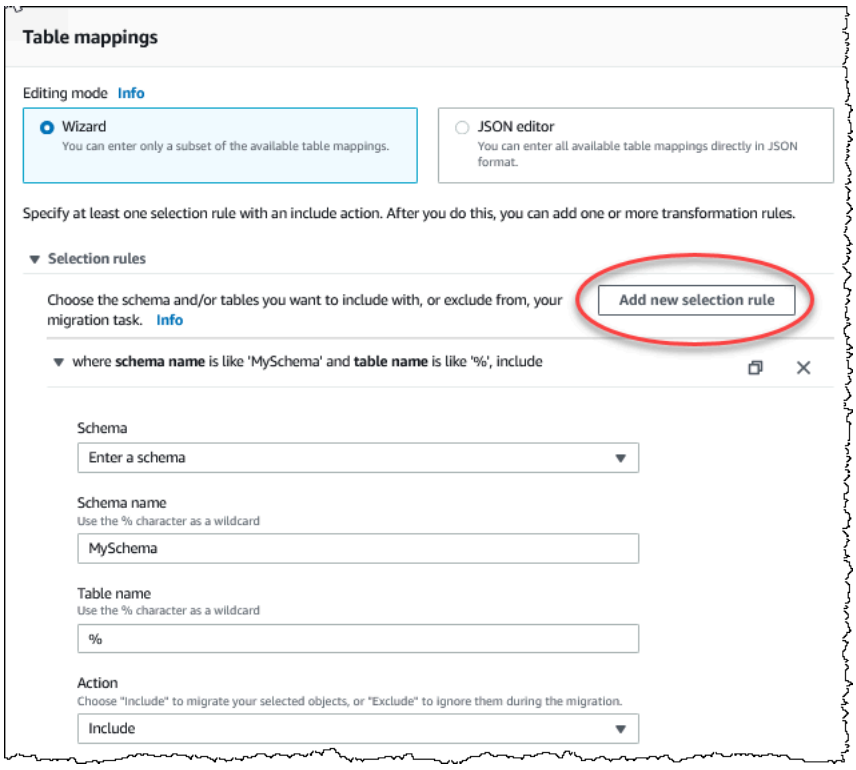
Pour spécifier une sélection de table, des critères de filtre et des transformations à l'aide de la console

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.

   Si vous êtes connecté en tant qu'utilisateur IAM, assurez-vous d'avoir les autorisations appropriées pour accéder à AWS DMS. Pour plus d'informations sur les autorisations requises, consultez Autorisations IAM nécessaires pour utiliser AWS DMS.
2. Sur la page Tableau de bord, choisissez Tâches de migration de base de données.
3. Choisissez Créer une tâche.
4. Dans la section Configuration de la tâche, entrez les informations relatives à la tâche, notamment l'Identifiant de tâche, l'Instance de réplication, le Point de terminaison de base de données source, le Point de terminaison de base de données cible et le Type de migration.



5. Dans la section Mappage de table, entrez le nom de schéma et le nom de table. Vous pouvez utiliser le signe « % » comme caractère générique lorsque vous spécifiez le nom de schéma ou le nom de table. Pour en savoir plus sur les autres caractères génériques que vous pouvez utiliser, consultez the section called "Caractères génériques dans le mappage de table". Spécifiez l'action à exécuter pour inclure ou exclure les données définies par le filtre.

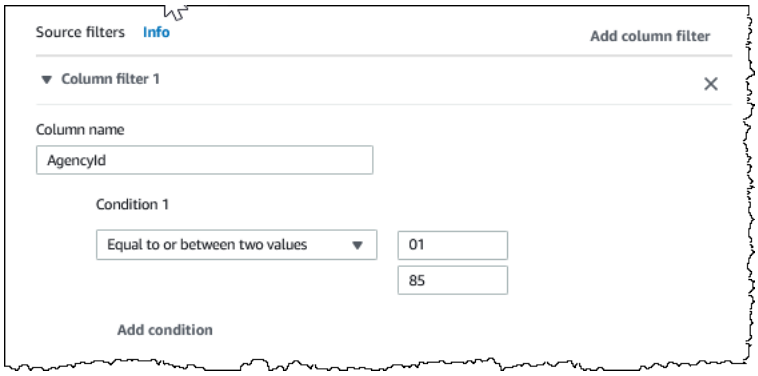


6. Spécifiez des informations de filtre à l'aide des liens Ajouter un filtre de colonne et Ajouter une condition.

   a. Choisissez Ajouter un filtre de colonne pour spécifier une colonne et des conditions.

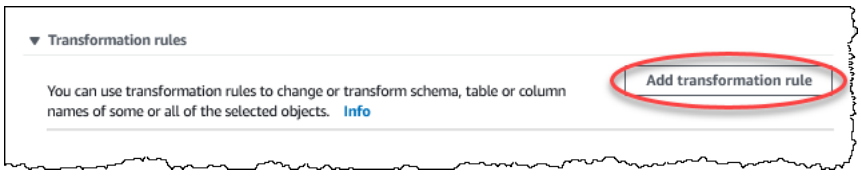
   b. Choisissez Ajouter une condition pour ajouter des conditions supplémentaires.

   L'exemple suivant montre un filtre pour la table **`Customers`** qui inclut les **`AgencyIDs`** entre **`01`** et **`85`**.

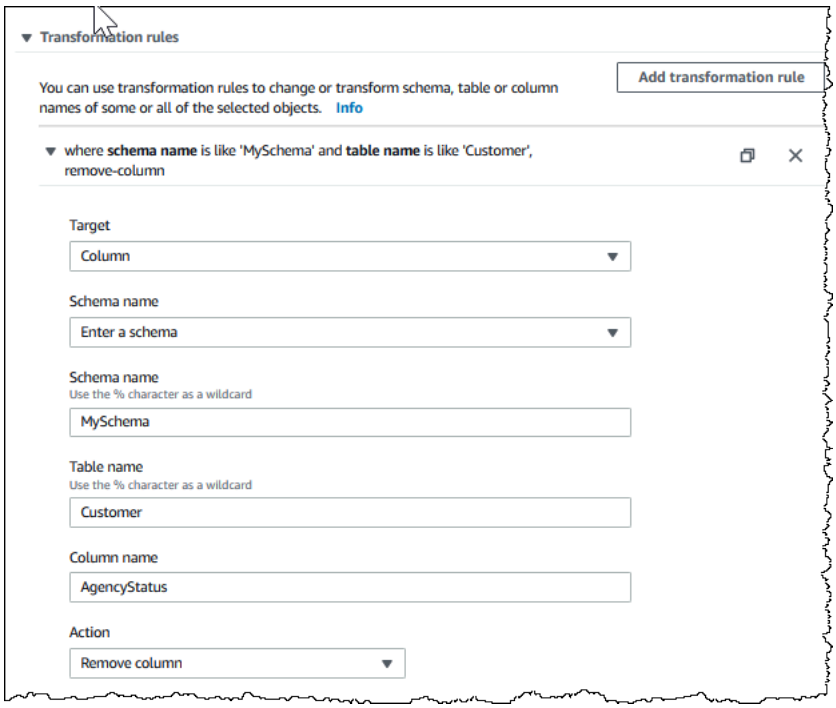


7. Lorsque vous avez créé vos sélections, choisissez Ajouter une nouvelle règle de sélection.

8. Une fois que vous avez créé au moins une règle de sélection, vous pouvez ajouter une transformation à la tâche. Choisissez Ajouter une règle de transformation.



9. Choisissez la cible que vous souhaitez transformer et saisissez les informations supplémentaires requises. L'exemple suivant montre une transformation qui supprime la colonne **`AgencyStatus`** de la table **`Customer`**.



10. Choisissez Ajouter une règle de transformation.
11. Choisissez Créer tâche.

> **Note**
>
> AWS DMS ne prend pas en charge plus d'une règle de transformation par niveau de schéma ou par niveau de table.

Spécification des règles de sélection de table et de transformation à l'aide de JSON

Pour spécifier les mappages de table à appliquer lors de la migration, vous pouvez créer un fichier JSON. Si vous créez une tâche de migration à l'aide de la console, vous pouvez rechercher ce

fichier JSON ou entrer le JSON directement dans la zone de mappage de table. Si vous utilisez l'interface de ligne de commande ou l'API pour effectuer des migrations, vous pouvez spécifier ce fichier en utilisant le paramètre `TableMappings` de l'opération d'API `CreateReplicationTask` ou `ModifyReplicationTask`.

AWS DMS ne peut traiter que des fichiers JSON de mappage de tables d'une taille maximale de 2 Mo. Nous vous recommandons de maintenir la taille de fichier JSON de la règle de mappage au-dessous de la limite de 2 Mo lorsque vous travaillez avec des tâches DMS. Cela permet d'éviter les erreurs inattendues lors de la création ou de la modification des tâches. Lorsqu'un fichier de règle de mappage dépasse la limite de 2 Mo, nous vous recommandons de répartir les tables entre plusieurs tâches afin que la taille du fichier de règle de mappage reste inférieure à cette limite.

Vous pouvez spécifier les tables, les vues et les schémas que vous souhaitez utiliser. Vous pouvez également effectuer des transformations de tables, de vues et de schémas, et spécifier des paramètres définissant la façon dont AWS DMS charge des tables et des vues individuelles. Vous créez des règles de mappage de table pour ces options à l'aide des types de règles suivants :

- Règles `selection` : identifiez les types et les noms des tables, des vues et des schémas sources à charger. Pour plus d'informations, consultez Règles et actions de sélection.
- Règles `transformation` : spécifiez certaines modifications ou certains ajouts apportés à des tables et des schémas sources particuliers dans la source avant qu'ils ne soient chargés dans la cible. Pour plus d'informations, consultez Règles et actions de transformation.

  En outre, pour définir le contenu des colonnes nouvelles et existantes, vous pouvez utiliser une expression dans une règle de transformation. Pour plus d'informations, consultez Utilisation d'expressions de règle de transformation pour définir le contenu d'une colonne.
- Règles `table-settings` : spécifiez la façon dont les tâches DMS chargent les données des tables individuelles. Pour plus d'informations, consultez Règles des paramètres de table et de collection et opérations.

> **Note**
>
> Pour les cibles Amazon S3, vous pouvez également baliser les objets S3 mappés aux tables et aux schémas sélectionnés à l'aide du type de règle `post-processing` et de l'action de règle `add-tag`. Pour plus d'informations, consultez Balisage d'objets Amazon S3.
> Pour les cibles suivantes, vous pouvez spécifier comment et où migrer les schémas et les tables sélectionnés vers la cible à l'aide du type de règle `object-mapping` :

- Amazon DynamoDB : pour plus d'informations, consultez Utilisation du mappage d'objet pour migrer les données vers DynamoDB.
- Amazon Kinesis : pour plus d'informations, consultez Utilisation du mappage d'objet pour migrer les données vers un flux de données Kinesis.
- Apache Kafka : pour plus d'informations, consultez Utilisation du mappage d'objet pour migrer les données vers une rubrique Kafka.

Règles et actions de sélection

Le mappage de tables vous permet de spécifier les tables, vues ou schémas que vous voulez utiliser grâce aux actions et règles de sélection. Pour les règles de mappage de tables qui utilisent le type de règle de sélection, vous pouvez appliquer les valeurs suivantes :

Paramètre	Valeurs possibles	Description
`rule-type`	`selection`	Règle de sélection. Définissez au moins une règle de sélection lorsque vous spécifiez un mapping de table.
`rule-id`	Valeur numérique.	Valeur numérique unique pour identifier la règle.
`rule-name`	Valeur alphanumérique.	Nom unique pour identifier la règle.
`rule-action`	`include`, `exclude`, `explicit`	Valeur qui inclut ou exclut le ou les objets sélectionnés par la règle. Si `explicit` est spécifié, vous pouvez sélectionner et inclure un seul objet correspondant à une table et un schéma explicitement spécifiés.
`object-locator`	Objet avec les paramètres suivants : • `schema-name` : nom du schéma. • `table-name` : nom de la table.	Nom de chaque schéma et de chaque table ou vue auxquels la règle s'applique. Vous pouvez également spécifier si une règle inclut uniquement des tables,

Paramètre	Valeurs possibles	Description
	• (Facultatif) `table-type` : `table | view | all`, pour indiquer si `table-name` fait uniquement référence aux tables, aux vues ou aux deux à la fois. L'argument par défaut est `table`. AWS DMS charge les vues uniquement dans le cadre d'une tâche à chargement complet. Si vous n'avez que des tâches de chargement complet et de capture des données modifiées (CDC), configurez au moins une full-load-only tâche pour charger vos vues. Les points de terminaison cibles n'acceptent pas tous les points de terminaison comme source de réplication, même à pleine charge (Amazon OpenSearch Service, par exemple). Vérifiez les limitations du point de terminaison cible.	uniquement des vues ou à la fois des tables et des vues. Si `rule-action` a pour valeur `include` ou `exclude`, vous pouvez utiliser le signe « % » comme caractère générique pour tout ou partie de la valeur des paramètres `schema-name` et `table-name`. Pour en savoir plus sur les autres caractères génériques que vous pouvez utiliser, consultez the section called "Caractères génériques dans le mappage de table". Ainsi, vous pouvez faire correspondre ces éléments : • Une table, une vue ou une collection unique dans un seul schéma • Une table, une vue ou une collection unique dans tout ou partie des schémas • Certaines ou toutes les tables et vues dans un schéma unique, ou les collections dans une seule base de données • Certaines ou toutes les tables et vues dans tout ou partie des schémas, ou les collections dans tout ou partie des bases de données Si `rule-action` a pour valeur `explicit`, vous pouvez uniquement spécifier le nom exact d'une table ou

Paramètre	Valeurs possibles	Description
		d'une vue unique et de son schéma (sans caractères génériques). Les sources prises en charge pour les vues sont les suivantes : • Oracle • Microsoft SQL Server • PostgreSQL • IBM Db2 LUW • IBM DB2 z/OS • SAP Adaptive Server Enterprise (ASE) • MySQL • AURORA • AURORA sans serveur • MariaDB **Note** AWS DMS ne charge jamais une vue source dans une vue cible. Une vue source est chargée dans une table équivalente de la cible avec le même nom que la vue dans la source. Les sources prises en charge pour les bases de données contenant des collections sont les suivantes : • MongoDB

Paramètre	Valeurs possibles	Description
		• Amazon DocumentDB
`load-order`	Nombre entier positif. La valeur maximale est 2 147 483 647.	Priorité pour le chargement des tables et des vues. Les tables et les vues avec des valeurs plus élevées sont chargées en premier.
`filters`	Tableau d'objets .	Un ou plusieurs objets pour filtrer la source. Vous spécifiez des paramètres d'objet pour filtrer sur une seule colonne de la source. Vous spécifiez plusieurs objets pour filtrer sur plusieurs colonnes. Pour plus d'informations, consultez Utilisation de filtres de source.

Example Migrer toutes les tables dans un schéma

L'exemple suivant migre toutes les tables d'un schéma nommé `Test` de la source vers votre point de terminaison cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            },
            "rule-action": "include"
        }
    ]
}
```

Example Migrer certaines tables d'un schéma

L'exemple suivant migre toutes les tables, sauf celles commençant par `DMS`, d'un schéma nommé `Test` de la source vers votre point de terminaison cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "selection",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "DMS%"
            },
            "rule-action": "exclude"
        }
    ]
}
```

Example Migration d'une table unique spécifiée dans un schéma unique

L'exemple suivant migre la table `Customer` du schéma `NewCust` de votre source vers votre point de terminaison cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "NewCust",
```

```
                "table-name": "Customer"
            },
            "rule-action": "explicit"
        }
    ]
}
```

> **Note**
>
> Vous pouvez explicitement procéder à une sélection dans plusieurs tables et schémas en spécifiant plusieurs règles de sélection.

Example Migration des tables dans un ordre défini

L'exemple suivant migre deux tables. La table `loadfirst` (avec priorité 1) est initialisée avant la table. `loadsecond`

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "loadsecond"
            },
            "rule-action": "include",
            "load-order": "2"
        },
        {
            "rule-type": "selection",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "loadfirst"
            },
            "rule-action": "include",
            "load-order": "1"
        }
```

```
    ]
}
```

> **Note**
>
> `load-order` s'applique à l'initialisation de la table. Le chargement d'une table successive n'attendra pas la fin du chargement d'une table précédente si `MaxFullLoadSubTasks` est supérieur à 1.

Example Migrer certaines vues d'un schéma

L'exemple suivant migre certaines vues d'un schéma nommé `Test` de votre source vers des tables équivalentes dans votre cible.

```
{
   "rules": [
        {
           "rule-type": "selection",
           "rule-id": "2",
           "rule-name": "2",
           "object-locator": {
               "schema-name": "Test",
               "table-name": "view_DMS%",
               "table-type": "view"
            },
           "rule-action": "include"
        }
    ]
}
```

Example Migrer toutes les tables et vues d'un schéma

L'exemple suivant migre toutes les tables et vues d'un schéma nommé `report` de votre source vers des tables équivalentes dans votre cible.

```
{
   "rules": [
        {
           "rule-type": "selection",
           "rule-id": "3",
           "rule-name": "3",
```

```
            "object-locator": {
                "schema-name": "report",
                "table-name": "%",
                "table-type": "all"
             },
            "rule-action": "include"
        }
    ]
}
```

Caractères génériques dans le mappage de table

Cette section décrit les caractères génériques que vous pouvez utiliser lorsque vous spécifiez les noms de schéma et de table pour le mappage de table.

Caractère générique	Correspondance
%	Zero or more characters
_	A single character
[_]	A literal underscore character
[ab]	A set of characters. For example, [ab] matches either 'a' or 'b'.
[a-d]	A range of characters. For example,[a-d] matches either 'a', 'b', 'c', or 'd'.

Pour les points de terminaison sources et cibles Oracle, vous pouvez utiliser l'attribut de connexion supplémentaire `escapeCharacter` pour spécifier un caractère d'échappement. Un caractère d'échappement vous permet d'utiliser un caractère générique spécifique dans les expressions comme s'il ne s'agissait pas d'un caractère générique. Par exemple, `escapeCharacter=#` vous permet d'utiliser « # » pour faire en sorte qu'un caractère générique agisse comme un caractère ordinaire dans une expression, comme dans l'exemple de code suivant.

```
{
```

```
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "542485267",
            "rule-name": "542485267",
            "object-locator": { "schema-name": "R00T", "table-name": "TEST#_T%" },
            "rule-action": "include",
            "filters": []
        }
    ]
}
```

Ici, le caractère d'échappement « # » fait en sorte que le caractère générique « _ » agisse comme un caractère normal. AWS DMS sélectionne les tables dans le schéma nomméR00T, chaque table ayant un nom avec TEST_T comme préfixe.

Règles et actions de transformation

Les actions de transformation permettent de spécifier les transformations que vous souhaitez appliquer au schéma, à la table ou à la vue sélectionné(e). Les règles de transformation sont facultatives.

Limites

- Vous ne pouvez pas appliquer plusieurs actions de règle de transformation au même objet (schéma, table, colonne, table-tablespace ou index-tablespace). Vous pouvez appliquer plusieurs actions de règle de transformation à n'importe quel niveau, à condition que chaque action de transformation soit appliquée à un objet différent.
- Les noms de table et de colonne sont sensibles à la casse dans les règles de transformation. Par exemple, vous devez fournir les noms de table et de colonne d'une base de données Oracle ou Db2 en majuscules.
- Les transformations ne sont pas prises en charge pour les noms de colonne rédigés de droite à gauche.
- Les transformations ne peuvent pas être effectuées sur des colonnes dont le nom contient des caractères spéciaux (par exemple, #, \, /, -).
- La seule transformation prise en charge pour les colonnes mappées aux types de données BLOB/CLOB consiste à supprimer la colonne sur la cible.
- AWS DMS ne prend pas en charge la réplication de deux tables sources vers une seule table cible. AWS DMS réplique les enregistrements de table en table et de colonne en colonne, conformément

aux règles de transformation de la tâche de réplication. Les noms des objets doivent être uniques pour éviter tout chevauchement.

Par exemple, une table source possède une colonne nommée `ID` et la table cible correspondante possède une colonne préexistante appelée `id`. Si une règle utilise une instruction `ADD-COLUMN` pour ajouter une nouvelle colonne appelée `id`, et une instruction SQLite pour remplir la colonne avec des valeurs personnalisées, un objet ambigu en double nommé `id` est créé et il n'est pas pris en charge.

Valeurs

Pour les règles de mappage de tables qui utilisent le type de règle de transformation, vous pouvez appliquer les valeurs suivantes :

Paramètre	Valeurs possibles	Description
`rule-type`	`transformation`	Valeur qui applique la règle à chaque objet spécifié par la règle de sélection. Utilisez, `transformation` sauf indication contraire.
`rule-id`	Valeur numérique.	Valeur numérique unique pour identifier la règle. Si vous spécifiez plusieurs règles de transformation pour le même objet (schéma, table, colonne, espace inter-tables ou espace table d'index), AWS DMS applique la règle de transformation avec le rule-id inférieur.
`rule-name`	Valeur alphanumérique.	Nom unique pour identifier la règle.
`object-locator`	Objet avec les paramètres suivants : • `schema-name` : nom du schéma. Pour les points de terminaison MongoDB et Amazon DocumentDB, il s'agit du nom de la base de	Nom de chaque schéma, table ou vue, espace de table, espace de table d'index et colonne auxquels la règle s'applique. Vous pouvez utiliser le signe « % » comme caractère générique pour tout ou partie de

Paramètre	Valeurs possibles	Description
	données contenant un ensemble de collections. • `table-name` : nom de la table, de la vue ou de la collection. • `table-tablespace-name` : nom d'un espace de table de table existant. • `index-tablespace-name` : nom d'un espace de table d'index existant. • `column-name` : nom d'une colonne existante. • `data-type` : nom d'un type de données de colonne existant.	la valeur de chaque paramètre `object-locator`, sauf `data-type`. Ainsi, vous pouvez faire correspondre ces éléments : • Une table ou une vue unique dans un seul schéma • Une table ou une vue unique dans tout ou partie des schémas • Certaines tables ou vues, ou toutes les tables et vues, dans un seul schéma • Certaines tables ou vues, ou toutes les tables et vues, dans tout ou partie des schémas • Une ou plusieurs colonnes dans la ou les table(s), la ou les vue(s) spécifiée(s) et le(s) schéma(s) spécifiés. • Les colonnes avec un `data-type` donné lorsque plusieurs colonnes sont spécifiées. Pour les valeurs possibles de `data-type`, consultez `data-type` décrit ci-dessous dans ce tableau. En outre, les paramètres `index-tablespace-name` ou `table-tablespace-name` sont disponibles uniquement pour correspondre à un point de terminaison source Oracle. Vous pouvez spécifier `table-tablespace-name` ou `index-tablespace-name` dans une règle,

Paramètre	Valeurs possibles	Description
		mais pas les deux. Vous pouvez donc établir une correspondance avec un des éléments suivants : • Un, plusieurs ou tous les espaces de table de table • Un, plusieurs ou tous les espaces de table d'index

Paramètre	Valeurs possibles	Description

`rule-action`	`add-column` , `include-column` , `remove-column`
	
	`rename`
	
	`convert-lowercase` , `convert-uppercase`
	
	`add-prefix` , `remove-prefix` , `replace-prefix`
	
	`add-suffix` , `remove-suffix` , `replace-suffix`
	
	`define-primary-key`
	
	`change-data-type`
	
	`add-before-image-columns`
		Transformation à appliquer à l'objet. Toutes les actions de règle de transformation sont sensibles à la casse.
		
		La valeur `add-column` du paramètre `rule-action` ajoute une colonne à une table. Vous ne pouvez toutefois pas ajouter une nouvelle colonne qui porte le même nom qu'une colonne existante de la même table.
		
		Lorsqu'elle est utilisée avec les paramètres `data-type` et `expression` , `add-column` spécifie la valeur des données de la nouvelle colonne.
		
		La valeur `change-data-type` pour `rule-action` est uniquemen t disponible pour les cibles de règle `column`.
		
		La valeur `include-column` du paramètre `rule-action` modifie le mode de la table pour supprimer toutes les colonnes par défaut et inclure les colonnes spécifiées. Plusieurs colonnes sont incluses dans la cible en invoquant la règle `include-column` plusieurs fois.
		
		Vous ne pouvez pas utiliser une règle `define-primary-key` lorsque

Paramètre	Valeurs possibles	Description
		le nom d'un schéma ou d'une table contient un caractère générique (%). Pour une tâche existante, les actions de la règle de transform ation qui modifient le schéma de la table cible telles que `remove-column` , `rename` ou `add-prefix` ne prendront effet qu'après le redémarrage de la tâche. Si vous reprenez la tâche après avoir ajouté la règle de transformation, vous pourriez constater un comportement inattendu de la colonne modifiée, notamment des données de colonne manquantes. Un redémarrage de la tâche est nécessaire pour garantir le bon fonctionnement de la règle de transformation.
`rule-target`	`schema`, `table`, `column`, `table-tablespace` , `index-tablespace`	Type d'objet que vous transformez. Les valeurs `table-tablespace` et `index-tablespace` ne sont disponibles que pour un point de terminaison cible Oracle. Assurez-vous de spécifier une valeur pour le paramètre que vous spécifiez dans le nom `object-locator` : `table-tablespace-name` ou `index-tablespace-name` .
`value`	Valeur alphanumérique qui suit les règles de dénomination pour le type de cible.	Nouvelle valeur pour les actions qui nécessitent une entrée, par exemple `rename`.

Paramètre	Valeurs possibles	Description
`old-value`	Valeur alphanumérique qui suit les règles de dénomination pour le type de cible.	Ancienne valeur pour les actions qui ont besoin d'être remplacées, par exemple `replace-prefix` .

Paramètre	Valeurs possibles	Description
`data-type`	`type` : type de données à utiliser si `rule-action` est `add-column` ou type de données de remplacement si `rule-action` est `change-data-type` . Ou, le nom du type de données de remplacement quand l'`rule-action` est `change-data-type` , la valeur de `column-name` est `"%"` et un paramètre `data-type` supplémentaire permettant d'identifier le type de données existant est inclus dans le `object-locator` . AWS DMS prend en charge les transformations de type de données de colonne pour les types de données DMS suivants : `"bytes", "date", "time", "datetime", "int1", "int2", "int4", "int8", "numeric", "real4", "real8", "string", "uint1", "uint2", "uint4", "uint8", "wstring", "blob", "nclob", "clob", "boolean", "set", "list" "map", "tuple"` `precision` : si la colonne ajoutée ou le type de données de remplacement a une précision, une valeur entière pour spécifier la précision. `scale` : si la colonne ajoutée ou le type de données de remplacement a	Exemple de paramètre `data-type` pour spécifier le type de données existant à remplacer. `{ "rules": [{ "rule-type": "selection", "rule-id": "1", "rule-name": "1", "object-locator": { "schema-name": "%", "table-name": "%" }, "rule-action": "include" }, { "rule-type": "transformation", "rule-id": "2", "rule-name": "2", "rule-target": "column", "object-locator": { "schema-name": "test", "table-name": "table_t", "column-name": "col10" }, "rule-action": "change-data-type", "data-type": { "type": "string", "length": "4092", "scale": "" } }] }`

Paramètre	Valeurs possibles	Description
	une échelle, une valeur entière ou de date/heure pour spécifier l'échelle. `length` : longueur de la nouvelle colonne de données (si utilisée avec `add-column`)	Ici, la colonne `col10` de la table `table_t` est remplacée par le type de données `string`.

Paramètre	Valeurs possibles	Description
`expression`	Valeur alphanumérique qui suit la syntaxe SQLite.	Lorsqu'il est utilisé avec l'`rule-action` définie sur `rename-schema`, le paramètre `expression` spécifie un nouveau schéma. Lorsqu'elle est utilisée avec l'`rule-action` définit sur `rename-table`, `expression` spécifie une nouvelle table. Lorsqu'elle est utilisée avec l'`rule-action` définie sur `rename-column`, `expression` spécifie une nouvelle valeur de nom de colonne. Lorsqu'elle est utilisée avec l'`rule-action` définie sur `add-column`, `expression` spécifie les données qui constituent une nouvelle colonne. Notez que seules les expressions sont prises en charge pour ce paramètre. Les opérateurs et les commandes ne sont pas pris en charge. Pour plus d'informations sur l'utilisation des expressions pour les règles de transformation, consultez Utilisation d'expressions de règle de transformation pour définir le contenu d'une colonne. Pour plus d'informations sur les expressions SQLite, consultez Utilisation des fonctions SQLite pour créer des expressions.

Paramètre	Valeurs possibles	Description
`primary-key-def`	Objet avec les paramètres suivants : • `name` : nom d'une nouvelle clé primaire ou d'un nouvel index unique pour la table ou la vue. • (Facultatif) `origin` : type de clé unique à définir : `primary-key` (valeur par défaut) ou `unique-index` . • `columns` : tableau de chaînes répertoriant les noms des colonnes selon l'ordre dans lequel ils apparaissent dans la clé primaire ou dans l'index unique.	Ce paramètre peut définir le nom, le type et le contenu d'une clé unique dans la table ou la vue transformée. Il le fait lorsque l'`rule-action` est définie sur `define-primary-key` et la `rule-target` est définie sur `table`. Par défaut, la clé unique est définie en tant que clé primaire.

Paramètre	Valeurs possibles	Description
`before-image-def`	Objet avec les paramètres suivants : • `column-prefix` : valeur ajoutée au début d'un nom de colonne. La valeur par défaut est `BI_`. • `column-suffix` : valeur ajoutée à la fin du nom de colonne. La valeur par défaut est vide. • `column-filter` : nécessite l'une des valeurs suivantes : `pk-only` (par défaut), `non-lob` (facultative) et `all` (facultative).	Ce paramètre définit une convention de dénomination pour identifier les colonnes d'image antérieure et spécifie un filtre pour identifier les colonnes source qui peuvent avoir des colonnes d'image antérieure créées pour elles sur la cible. Vous pouvez spécifier ce paramètre lorsque l'`rule-action` est définie sur `add-before-image-columns` et la `rule-target` est définie sur `column`. Ne définissez pas les deux `column-prefix` et `column-suffix` sur des chaînes vides. Pour `column-filter`, sélectionnez: • `pk-only` : pour ajouter uniquement les colonnes qui font partie des clés primaires de la table. • `non-lob` : pour ajouter uniquement les colonnes qui ne sont pas de type LOB. • `all` : pour ajouter une colonne ayant une valeur d'image antérieure. Pour plus d'informations sur la prise en charge de l'image antérieure pour les points de terminaison cibles AWS DMS, consultez :

Paramètre	Valeurs possibles	Description
		• Utilisation d'une image antérieure pour afficher les valeurs d'origine des lignes CDC pour un flux de données Kinesis en tant que cible • Utilisation d'une image antérieure pour afficher les valeurs originales des lignes de la CDC pour Apache Kafka comme cible

Exemples

Example Renommer un schéma

L'exemple suivant renomme un schéma `Test` de votre source en `Test1` dans votre cible.

```
{

    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "transformation",
            "rule-id": "2",
            "rule-name": "2",
            "rule-action": "rename",
            "rule-target": "schema",
            "object-locator": {
                "schema-name": "Test"
            },
            "value": "Test1"
        }
```

```
    ]
}
```

Example Renommer une table

L'exemple suivant renomme une table `Actor` de votre source en `Actor1` dans votre cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "transformation",
            "rule-id": "2",
            "rule-name": "2",
            "rule-action": "rename",
            "rule-target": "table",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "Actor"
            },
            "value": "Actor1"
        }
    ]
}
```

Example Renommer une colonne

L'exemple suivant renomme une colonne de la table `Actor` en remplaçant `first_name` dans votre source par `fname` dans votre cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
```

```
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
         {
            "rule-type": "transformation",
            "rule-id": "4",
            "rule-name": "4",
            "rule-action": "rename",
            "rule-target": "column",
            "object-locator": {
                "schema-name": "test",
                "table-name": "Actor",
                "column-name" : "first_name"
            },
            "value": "fname"
        }
    ]
}
```

Example Renommer un espace de table de table Oracle

L'exemple suivant renomme l'espace de table de table nommé `SetSpace` d'une table nommée `Actor` dans votre source Oracle en `SceneTblSpace` dans votre point de terminaison cible Oracle.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Play",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "transformation",
```

```
            "rule-id": "2",
            "rule-name": "2",
            "rule-action": "rename",
            "rule-target": "table-tablespace",
            "object-locator": {
                "schema-name": "Play",
                "table-name": "Actor",
                "table-tablespace-name": "SetSpace"
            },
            "value": "SceneTblSpace"
        }
    ]
}
```

Example Renommer un espace de table d'index Oracle

L'exemple suivant renomme l'espace de table d'index nommé `SetISpace` d'une table nommée `Actor` dans votre source Oracle en `SceneIdxSpace` dans votre point de terminaison cible Oracle.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Play",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "transformation",
            "rule-id": "2",
            "rule-name": "2",
            "rule-action": "rename",
            "rule-target": "table-tablespace",
            "object-locator": {
                "schema-name": "Play",
                "table-name": "Actor",
                "table-tablespace-name": "SetISpace"
            },
            "value": "SceneIdxSpace"
```

```
        }
    ]
}
```

Example Ajouter une colonne

Dans l'exemple suivant une colonne `datetime` est ajoutée à la table `Actor` du schéma `test`.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "transformation",
            "rule-id": "2",
            "rule-name": "2",
            "rule-action": "add-column",
            "rule-target": "column",
            "object-locator": {
                "schema-name": "test",
                "table-name": "actor"
            },
            "value": "last_updated",
            "data-type": {
                "type": "datetime",
                "precision": 6
            }
        }
    ]
}
```

Example Supprimer une colonne

L'exemple suivant transforme la table nommée `Actor` dans votre source pour en supprimer toutes les colonnes commençant par les caractères `col` dans votre cible.

```
{
  "rules": [{
  "rule-type": "selection",
  "rule-id": "1",
  "rule-name": "1",
  "object-locator": {
   "schema-name": "test",
   "table-name": "%"
  },
  "rule-action": "include"
 }, {
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "rule-action": "remove-column",
  "rule-target": "column",
  "object-locator": {
   "schema-name": "test",
   "table-name": "Actor",
   "column-name": "col%"
  }
 }]
 }
```

Example Convertir en minuscules

L'exemple suivant convertit un nom de table `ACTOR` dans votre source en `actor` dans votre cible.

```
{
 "rules": [{
  "rule-type": "selection",
  "rule-id": "1",
  "rule-name": "1",
  "object-locator": {
   "schema-name": "test",
   "table-name": "%"
  },
  "rule-action": "include"
 }, {
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "rule-action": "convert-lowercase",
```

```
  "rule-target": "table",
  "object-locator": {
   "schema-name": "test",
   "table-name": "ACTOR"
  }
 }]
}
```

Example Convertir en majuscules

L'exemple suivant convertit toutes les colonnes de toutes les tables et tous les schémas en les faisant passer de lettres minuscules dans votre source à des lettres majuscules dans votre cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "transformation",
            "rule-id": "2",
            "rule-name": "2",
            "rule-action": "convert-uppercase",
            "rule-target": "column",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%",
                "column-name": "%"
            }
        }
    ]
}
```

Example Ajouter un préfixe

L'exemple suivant transforme toutes les tables de votre source en leur ajoutant le préfixe `DMS_` dans votre cible.

```
{
  "rules": [{
  "rule-type": "selection",
  "rule-id": "1",
  "rule-name": "1",
  "object-locator": {
   "schema-name": "test",
   "table-name": "%"
  },
  "rule-action": "include"
 }, {
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "rule-action": "add-prefix",
  "rule-target": "table",
  "object-locator": {
   "schema-name": "test",
   "table-name": "%"
  },
  "value": "DMS_"
 }]

}
```

Example Remplacer un préfixe

L'exemple suivant transforme toutes les colonnes contenant le préfixe `Pre_` dans votre source en remplaçant le préfixe par `NewPre_` dans votre cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "test",
```

```
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "transformation",
            "rule-id": "2",
            "rule-name": "2",
            "rule-action": "replace-prefix",
            "rule-target": "column",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%",
                "column-name": "%"
            },
            "value": "NewPre_",
            "old-value": "Pre_"
        }
    ]
}
```

Example Supprimer un suffixe

L'exemple suivant transforme toutes les tables de votre source en leur supprimant le suffixe _DMS dans votre cible.

```
{
 "rules": [{
  "rule-type": "selection",
  "rule-id": "1",
  "rule-name": "1",
  "object-locator": {
   "schema-name": "test",
   "table-name": "%"
  },
  "rule-action": "include"
 }, {
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "rule-action": "remove-suffix",
  "rule-target": "table",
  "object-locator": {
```

```
   "schema-name": "test",
   "table-name": "%"
  },
  "value": "_DMS"
 }]
}
```

Example Définir une clé primaire

L'exemple suivant définit une clé primaire nommée `ITEM-primary-key` sur trois colonnes de la table `ITEM` migrée vers votre point de terminaison cible.

```
{
 "rules": [{
  "rule-type": "selection",
  "rule-id": "1",
  "rule-name": "1",
  "object-locator": {
   "schema-name": "inventory",
   "table-name": "%"
  },
  "rule-action": "include"
 }, {
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "rule-action": "define-primary-key",
  "rule-target": "table",
  "object-locator": {
   "schema-name": "inventory",
   "table-name": "ITEM"
  },
  "primary-key-def": {
   "name": "ITEM-primary-key",
   "columns": [
    "ITEM-NAME",
    "BOM-MODEL-NUM",
    "BOM-PART-NUM"
   ]
              }
 }]
}
```

Example Définir un index unique

L'exemple suivant définit un index unique nommé `ITEM-unique-idx` sur trois colonnes de la table `ITEM` migrée vers votre point de terminaison cible.

```
{
 "rules": [{
  "rule-type": "selection",
  "rule-id": "1",
  "rule-name": "1",
  "object-locator": {
   "schema-name": "inventory",
   "table-name": "%"
  },
  "rule-action": "include"
 }, {
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "rule-action": "define-primary-key",
  "rule-target": "table",
  "object-locator": {
   "schema-name": "inventory",
   "table-name": "ITEM"
  },
  "primary-key-def": {
   "name": "ITEM-unique-idx",
   "origin": "unique-index",
   "columns": [
    "ITEM-NAME",
    "BOM-MODEL-NUM",
    "BOM-PART-NUM"
   ]
              }
 }]
}
```

Example Modification du type de données de la colonne cible

L'exemple suivant remplace le type de données d'une colonne cible nommée `SALE_AMOUNT` à partir d'un type de données existant par `int8`.

```
{
```

```
    "rule-type": "transformation",
    "rule-id": "1",
    "rule-name": "RuleName 1",
    "rule-action": "change-data-type",
    "rule-target": "column",
    "object-locator": {
        "schema-name": "dbo",
        "table-name": "dms",
        "column-name": "SALE_AMOUNT"
    },
    "data-type": {
        "type": "int8"
    }
}
```

Example Ajouter une colonne d'image antérieure

Pour une colonne source appelée emp_no, la règle de transformation de l'exemple suivant ajoute une nouvelle colonne appelée BI_emp_no dans la cible.

```
{
 "rules": [{
   "rule-type": "selection",
   "rule-id": "1",
   "rule-name": "1",
   "object-locator": {
    "schema-name": "%",
    "table-name": "%"
   },
   "rule-action": "include"
  },
  {
   "rule-type": "transformation",
   "rule-id": "2",
   "rule-name": "2",
   "rule-target": "column",
   "object-locator": {
    "schema-name": "%",
    "table-name": "employees"
   },
   "rule-action": "add-before-image-columns",
   "before-image-def": {
    "column-prefix": "BI_",
```

```
    "column-suffix": "",
    "column-filter": "pk-only"
   }
  }
 ]
}
```

Ici, l'instruction suivante remplit une colonne `BI_emp_no` dans la ligne correspondante avec 1.

```
UPDATE employees SET emp_no = 3 WHERE BI_emp_no = 1;
```

Lorsque vous écrivez des mises à jour CDC pour AWS DMS les cibles prises en charge, la `BI_emp_no` colonne permet de savoir quelles lignes contiennent des valeurs mises à jour dans la `emp_no` colonne.

Utilisation d'expressions de règle de transformation pour définir le contenu d'une colonne

Pour définir le contenu de colonnes nouvelles et existantes, vous pouvez utiliser une expression dans une règle de transformation. Par exemple, des expressions peuvent vous permettre d'ajouter une colonne ou de répliquer des en-têtes de table source à une cible. Vous pouvez également utiliser des expressions pour signaler des enregistrements des tables cibles comme insérés, mis à jour ou supprimés à la source.

Rubriques

- Ajouter une colonne à l'aide d'une expression
- Signalement d'enregistrements cibles à l'aide d'une expression
- Réplication des en-têtes d'une table source à l'aide d'expressions
- Utilisation des fonctions SQLite pour créer des expressions
- Ajout de métadonnées à une table cible à l'aide d'expressions

Ajouter une colonne à l'aide d'une expression

Pour ajouter des colonnes à des tables à l'aide d'une expression dans une règle de transformation, utilisez une action de règle `add-column` et une cible de règle `column`.

L'exemple suivant ajoute une nouvelle colonne à la table `ITEM`. Il définit le nouveau nom de colonne sur `FULL_NAME`, avec un type de données `string` de 50 caractères. L'expression concatène

les valeurs de deux colonnes existantes, `FIRST_NAME` et `LAST_NAME`, pour les évaluer selon `FULL_NAME`. `schema-name`, `table-name` et les paramètres d'expression font référence aux objets de la table de base de données source. `Value` et le bloc `data-type` font référence aux objets de la table de base de données cible.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "transformation",
            "rule-id": "2",
            "rule-name": "2",
            "rule-action": "add-column",
            "rule-target": "column",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "ITEM"
            },
            "value": "FULL_NAME",
            "expression": "$FIRST_NAME||'_'||$LAST_NAME",
            "data-type": {
                 "type": "string",
                 "length": 50
            }
        }
    ]
}
```

Signalement d'enregistrements cibles à l'aide d'une expression

Pour signaler des enregistrements dans des tables cibles comme insérés, mis à jour ou supprimés dans la table source, utilisez une expression dans une règle de transformation. L'expression utilise une fonction `operation_indicator` pour signaler les enregistrements. Les enregistrements

supprimés de la source ne sont pas supprimés de la cible. Au lieu de cela, l'enregistrement cible est signalé avec une valeur fournie par l'utilisateur pour indiquer qu'il a été supprimé de la source.

> **Note**
>
> La fonction `operation_indicator` fonctionne uniquement sur les tables qui ont une clé primaire à la fois sur les bases de données sources et cibles.

Par exemple, la règle de transformation suivante ajoute d'abord une nouvelle colonne `Operation` à une table cible. Elle met ensuite à jour la colonne avec la valeur `D` chaque fois qu'un enregistrement est supprimé d'une table source.

```
{
      "rule-type": "transformation",
      "rule-id": "2",
      "rule-name": "2",
      "rule-target": "column",
      "object-locator": {
        "schema-name": "%",
        "table-name": "%"
      },
      "rule-action": "add-column",
      "value": "Operation",
      "expression": "operation_indicator('D', 'U', 'I')",
      "data-type": {
        "type": "string",
        "length": 50
      }
}
```

Réplication des en-têtes d'une table source à l'aide d'expressions

Par défaut, les en-têtes des tables sources ne sont pas répliqués sur la cible. Pour indiquer les en-têtes à répliquer, utilisez une règle de transformation avec une expression qui inclut l'en-tête de colonne du tableau.

Vous pouvez utiliser les en-têtes de colonne suivants dans les expressions.

En-tête	Valeur de la réplication continue	Valeur en pleine charge	Type de données
AR_H_STREAM_POSITION	Valeur de position du flux à partir de la source. Cette valeur peut être le numéro de modification système (SCN) ou le numéro de séquence de journal (LSN), selon le point de terminaison source.	Une chaîne vide.	CHAÎNE
AR_H_TIMESTAMP	Horodatage indiquant l'heure de la modification.	Horodatage indiquant l'heure à laquelle les données arrivent au niveau de la cible.	DATETIME (scale=7)
AR_H_COMMIT_TIMESTAMP	Un horodatage indiquant l'heure de la validation.	Horodatage indiquant l'heure actuelle.	DATETIME (scale=7)
AR_H_OPERATION	INSERT, UPDATE ou DELETE	INSERT	CHAÎNE
AR_H_USER	Nom d'utilisateur, ID ou toute autre information fournie par la source sur l'utilisateur qui a effectué la modification. Cet en-tête est pris en charge uniquement sur les points de terminaison source	Transformation à appliquer à l'objet. Les actions de règle de transformation sont sensibles à la casse.	CHAÎNE

En-tête	Valeur de la réplication continue	Valeur en pleine charge	Type de données
	SQL Server et Oracle (version 11.2.0.3 et supérieure).		
AR_H_CHANGE_SEQ	Nombre incrémentiel unique issu de la base de données source, composé d'un horodatage et d'un nombre incrémentiel automatique. La valeur dépend du système de base de données source.	Une chaîne vide.	CHAÎNE

Dans l'exemple suivant, une nouvelle colonne est ajoutée à la cible en utilisant la valeur de position du flux à partir de la source. Pour SQL Server, la valeur de position de flux est le LSN du point de terminaison source. Pour Oracle, la valeur de position de flux est le SCN du point de terminaison source.

```
{
      "rule-type": "transformation",
     "rule-id": "2",
      "rule-name": "2",
      "rule-target": "column",
      "object-locator": {
        "schema-name": "%",
        "table-name": "%"
      },
      "rule-action": "add-column",
      "value": "transact_id",
      "expression": "$AR_H_STREAM_POSITION",
      "data-type": {
        "type": "string",
        "length": 50
      }
```

```
    }
```

L'exemple suivant ajoute une nouvelle colonne à la cible qui possède un nombre incrémentiel unique issu de la source. Cette valeur représente un nombre unique à 35 chiffres au niveau de la tâche. Les 16 premiers chiffres font partie d'un horodatage et les 19 derniers chiffres sont le numéro record_id incrémenté par le SGBD.

```
{
"rule-type": "transformation",
"rule-id": "2",
"rule-name": "2",
"rule-target": "column",
"object-locator": {
"schema-name": "%",
"table-name": "%"
},
"rule-action": "add-column",
"value": "transact_id",
"expression": "$AR_H_CHANGE_SEQ",
"data-type": {
"type": "string",
"length": 50
}
}
```

Utilisation des fonctions SQLite pour créer des expressions

Vous utilisez les paramètres de table pour spécifier les paramètres que vous souhaitez appliquer à la table ou à la vue sélectionnée pour une opération spécifiée. Les règles table-settings sont facultatives.

> **Note**
>
> Plutôt que d'utiliser des tables et des vues, les bases de données MongoDB et DocumentDB stockent les enregistrements de données sous forme de documents regroupés dans des collections. Ainsi, lors de la migration depuis une source MongoDB ou DocumentDB, prenez en compte le type de segmentation par plage des paramètres de chargement parallèle pour les collections sélectionnées plutôt que les tables et les vues.

Rubriques

- Utilisation d'une expression CASE
- Exemples

Vous trouverez ci-dessous les fonctions de chaîne que vous pouvez utiliser pour créer des expressions de règle de transformation.

Fonctions de chaîne	Description
`lower(x)`	La fonction `lower(x)` renvoie une copie de la chaîne *x* avec tous les caractères convertis en minuscules. La fonction `lower` intégrée par défaut fonctionne uniquement pour les caractères ASCII.
`upper(x)`	La fonction `upper(x)` renvoie une copie de la chaîne *x* avec tous les caractères convertis en majuscules. La fonction `upper` intégrée par défaut fonctionne uniquement pour les caractères ASCII.
`ltrim(x,y)`	La fonction `ltrim(x,y)` renvoie une chaîne constituée en supprimant du côté gauche de x tous les caractères qui apparaissent dans y. S'il n'y a aucune valeur pour y, `ltrim(x)` supprime les espaces du côté gauche de x.
`replace(x,y,z)`	La fonction `replace(x,y,z)` renvoie une chaîne constituée en substituant la chaîne z pour chaque occurrence de la chaîne y dans la chaîne x.
`rtrim(x,y)`	La fonction `rtrim(x,y)` renvoie une chaîne constituée en supprimant du côté droit de x tous les caractères qui apparaiss ent dans y. S'il n'y a aucune valeur pour y, `rtrim(x)` supprime les espaces du côté droit de x.
`substr(x,y,z)`	La fonction `substr(x,y,z)` renvoie une sous-chaîne de la chaîne d'entrée *x* qui commence par le *y*ème caractère et qui comporte *z* caractères.

Fonctions de chaîne	Description
	Si *z* est omis, `substr(x,y)` renvoie tous les caractères jusqu'à la fin de la chaîne *x* en commençant par le *y*ème caractère. Le caractère le plus à gauche de *x* est le 1er caractère. Si *y* est négatif, le premier caractère de la sous-chaîne est trouvé en comptant à partir de la droite plutôt que de la gauche. Si *z* est négatif, les `abs(z)` caractères précédant le *y*ème caractère sont renvoyés. Si *x* est une chaîne, les index des caractères font référence aux caractères UTF-8 réels. Si *x* est un objet BLOB, les index font référence à des octets.
`trim(x,y)`	La fonction `trim(x,y)` renvoie une chaîne constituée en supprimant des deux côtés de *x* tous les caractères qui apparaissent dans *y*. Si *y* n'a pas de valeur, `trim(x)` supprime les espaces des deux côtés de *x*.

Vous trouverez ci-dessous les fonctions LOB que vous pouvez utiliser pour créer des expressions de règle de transformation.

Fonctions LOB	Description
`hex(x)`	La fonction `hex` reçoit un objet BLOB en tant qu'argument et renvoie une version en chaîne hexadécimale en majuscules du contenu BLOB.
`randomblob ( N)`	La fonction `randomblob( N)` renvoie un objet BLOB de *N* octet(s) contenant des octets pseudo-aléatoires. Si *N* est inférieur à 1, un objet BLOB aléatoire de 1 octet est renvoyé.
`zeroblob(N)`	La fonction `zeroblob( N)` renvoie un objet BLOB composé de *N* octet(s) de 0x00.

Vous trouverez ci-dessous les fonctions numériques que vous pouvez utiliser pour créer des expressions de règle de transformation.

Fonctions numériques	Description
`abs(x)`	La fonction `abs(x)` renvoie la valeur absolue de l'argument numérique *x*. La fonction `abs(x)` renvoie NULL si *x* est NULL. La fonction `abs(x)` renvoie 0.0 si *x* est une chaîne ou un objet BLOB qui ne peut pas être converti(e) en valeur numérique.
`random()`	La fonction `random` renvoie un entier pseudo-al éatoire compris entre -9 223 372 036 854 775 808 et +9 223 372 036 854 775 807.
`round (x,y)`	La fonction `round (x,y)` renvoie une valeur à virgule flottante *x* arrondie à *y* chiffres à droite de la virgule décimale. Si *y* n'a pas de valeur, elle est supposée être 0.
`max (x,y...)`	La fonction `max` à plusieurs arguments renvoie l'argument avec la valeur maximale, ou renvoie NULL si l'un des arguments est NULL. La fonction `max` recherche dans ses arguments de gauche à droite un argument qui définit une fonction de classement. Si elle en trouve un, elle utilise cette fonction de classement pour toutes les comparaisons de chaînes. Si aucun des arguments de `max` ne définit une fonction de classement, la fonction de classement `BINARY` est utilisée. La fonction `max` est une fonction simple lorsqu'elle possède deux arguments ou plus, mais elle fonctionne comme une fonction d'agrégation si elle n'a qu'un seul argument.
`min (x,y...)`	La fonction `min` à plusieurs arguments renvoie l'argument avec la valeur minimale. La fonction `min` recherche dans ses arguments de gauche à droite un argument qui définit une fonction de classement. Si elle en trouve un, elle utilise cette fonction de classement pour toutes les comparaisons de chaînes. Si aucun des arguments de `min` ne définit une fonction de classement, la fonction de classement `BINARY` est utilisée. La fonction `min` est une

Fonctions numériques	Description
	fonction simple lorsqu'elle possède deux arguments ou plus, mais elle fonctionne comme une fonction d'agrégation si elle n'a qu'un seul argument.

Vous trouverez ci-dessous les fonctions de vérification NULL que vous pouvez utiliser pour créer des expressions de règle de transformation.

Fonctions de vérification NULL	Description
`coalesce (x,y...)`	La fonction `coalesce` renvoie une copie de son premier argument non NULL, mais elle renvoie NULL si tous les arguments sont NULL. La fonction coalesce comporte au moins deux arguments.
`ifnull(x,y)`	La fonction `ifnull` renvoie une copie de son premier argument non NULL, mais elle renvoie NULL si les deux arguments sont NULL. La fonction `ifnull` possède exactement deux arguments. La fonction `ifnull` est identique à `coalesce` avec deux arguments.
`nullif(x,y)`	La fonction `nullif(x,y)` renvoie une copie de son premier argument si les arguments sont différents, mais elle renvoie NULL si les arguments sont identiques. La fonction `nullif(x,y)` recherche dans ses arguments de gauche à droite un argument qui définit une fonction de classement. Si elle en trouve un, elle utilise cette fonction de classement pour toutes les comparaisons de chaînes. Si aucun des arguments NULLIF ne définit de fonction de classement, la fonction de classement `BINARY` est utilisée.

Vous trouverez ci-dessous les fonctions de date/heure que vous pouvez utiliser pour créer des expressions de règle de transformation.

Fonctions de date et d'heure	Description
`date(`*`timestring`* `, `*`modifier`*`, `*`modifier`*`...)`	La fonction `date` renvoie la date au format AAAA-MM-JJ.
`time(`*`timestring`* `, `*`modifier`*`, `*`modifier`*`...)`	La fonction `time` renvoie l'heure au format HH:MM:SS.
`datetime(` *`timestring`* `, `*`modifier`*`, `*`modifier`*`...)`	La fonction `datetime` renvoie la date et l'heure au format AAAA-MM-JJ HH:MM:SS.
`julianday(` *`timestrin g`* `, `*`modifier`*`, `*`modifier`*`...)`	La fonction `julianday` renvoie le nombre de jours écoulés depuis midi, heure de Greenwich, le 24 novembre 4714 av JC.
`strftime(` *`format`*`, `*`timestring`* `, `*`modifier`*`, `*`modifier`*`...)`	La fonction `strftime` renvoie la date conformément à la chaîne de format spécifiée comme premier argument, en utilisant l'une des variables suivantes : `%d` : jour du mois `%H` : heure entre 00 et 24 `%f` :** fractions de seconde SS.SSS `%j` : jour de l'année entre 001 et 366 `%J` :** numéro du jour julien `%m` : mois entre 01 et 12 `%M` : minute entre 00 et 59 `%s` : secondes depuis le 01/01/1970 `%S` : secondes entre 00 et 59 `%w` : jour de la semaine entre 0 et 6, dimanche = 0 `%W` : semaine de l'année entre 00 et 53

Fonctions de date et d'heure	Description
	%Y : année entre 0000 et 9999 %%: %

Vous trouverez ci-dessous une fonction de hachage que vous pouvez utiliser pour créer des expressions de règle de transformation.

Fonction de hachage	Description
`hash_sha256( x)`	La fonction `hash` génère une valeur de hachage pour une colonne d'entrée (à l'aide de l'algorithme SHA-256) et renvoie la valeur hexadécimale de la valeur de hachage générée. Pour utiliser la fonction `hash` dans une expression, ajoutez `hash_sha256( x)` à l'expression et remplacez *x* par le nom de la colonne source.

Utilisation d'une expression CASE

L'expression `CASE` SQLite évalue une liste de conditions et renvoie une expression en fonction du résultat. La syntaxe est la suivante.

```
    CASE case_expression
     WHEN when_expression_1 THEN result_1
     WHEN when_expression_2 THEN result_2
     ...
     [ ELSE result_else ]
    END

# Or

     CASE
     WHEN case_expression THEN result_1
     WHEN case_expression THEN result_2
     ...
     [ ELSE result_else ]
    END
```

Exemples

Example d'ajout d'une nouvelle colonne de chaîne à la table cible en utilisant une condition de casse

L'exemple de règle de transformation suivant ajoute une nouvelle colonne de chaîne `emp_seniority` à la table cible `employee`. Il utilise la fonction `round` SQLite sur la colonne de salaire, avec une condition de casse pour vérifier si le salaire est supérieur ou égal à 20 000. Si c'est le cas, la colonne obtient la valeur `SENIOR` et tout le reste a la valeur `JUNIOR`.

```
 {
     "rule-type": "transformation",
     "rule-id": "2",
     "rule-name": "2",
     "rule-action": "add-column",
     "rule-target": "column",
     "object-locator": {
       "schema-name": "public",
       "table-name": "employee"
     },
     "value": "emp_seniority",
     "expression": " CASE WHEN round($emp_salary)>=20000 THEN 'SENIOR' ELSE 'JUNIOR'
END",
     "data-type": {
       "type": "string",
       "length": 50
     }

 }
```

Example d'ajout d'une nouvelle colonne de date à la table cible

L'exemple suivant ajoute une nouvelle colonne de date `createdate` à la table cible `employee`. Lorsque vous utilisez la fonction de date SQLite `datetime`, la date est ajoutée à la table récemment créée pour chaque ligne insérée.

```
 {
     "rule-type": "transformation",
     "rule-id": "2",
     "rule-name": "2",
     "rule-action": "add-column",
     "rule-target": "column",
     "object-locator": {
       "schema-name": "public",
```

```
        "table-name": "employee"
      },
      "value": "createdate",
      "expression": "datetime ()",
      "data-type": {
        "type": "datetime",
        "precision": 6
      }
  }
```

Example d'ajout d'une nouvelle colonne numérique à la table cible

L'exemple suivant ajoute une nouvelle colonne numérique `rounded_emp_salary` à la table cible `employee`. Il utilise la fonction `round` SQLite pour ajouter le salaire arrondi.

```
  {
      "rule-type": "transformation",
      "rule-id": "2",
      "rule-name": "2",
      "rule-action": "add-column",
      "rule-target": "column",
      "object-locator": {
        "schema-name": "public",
        "table-name": "employee"
      },
      "value": "rounded_emp_salary",
      "expression": "round($emp_salary)",
      "data-type": {
        "type": "int8"
      }
  }
```

Example d'ajout d'une nouvelle colonne de chaîne à la table cible en utilisant la fonction de hachage

L'exemple suivant ajoute une nouvelle colonne de chaîne `hashed_emp_number` à la table cible `employee`. La fonction `hash_sha256(`*x*`)` SQLite crée des valeurs hachées sur la cible pour la colonne source `emp_number`.

```
  {
      "rule-type": "transformation",
      "rule-id": "2",
      "rule-name": "2",
      "rule-action": "add-column",
```

```
    "rule-target": "column",
    "object-locator": {
      "schema-name": "public",
      "table-name": "employee"
    },
    "value": "hashed_emp_number",
    "expression": "hash_sha256($emp_number)",
    "data-type": {
      "type": "string",
      "length": 64
    }
}
```

Ajout de métadonnées à une table cible à l'aide d'expressions

Vous pouvez ajouter les informations de métadonnées à la table cible en utilisant les expressions suivantes :

- `$AR_M_SOURCE_SCHEMA` : nom du schéma source.
- `$AR_M_SOURCE_TABLE_NAME` : nom de la table source.
- `$AR_M_SOURCE_COLUMN_NAME` : nom d'une colonne de la table source.
- `$AR_M_SOURCE_COLUMN_DATATYPE` : type de données d'une colonne de la table source.

Example d'ajout d'une colonne pour un nom de schéma en utilisant le nom de schéma issu de la source

L'exemple suivant ajoute une nouvelle colonne nommée `schema_name` à la cible en utilisant le nom de schéma issu de la source.

```
{
    "rule-type": "transformation",
    "rule-id": "2",
    "rule-name": "2",
    "rule-action": "add-column",
    "rule-target": "column",
    "object-locator": {
      "schema-name": "%",
      "table-name": "%"
    },
    "rule-action": "add-column",
    "value":"schema_name",
```

```
      "expression": "$AR_M_SOURCE_SCHEMA",
      "data-type": {
         "type": "string",
         "length": 50
      }
  }
```

Règles des paramètres de table et de collection et opérations

Utilisez les paramètres de table pour spécifier les paramètres que vous souhaitez appliquer à une table ou une vue sélectionnée pour une opération spécifiée. Les règles des paramètres de table sont facultatives, en fonction des exigences en matière de point de terminaison et de migration.

Plutôt que d'utiliser des tables et des vues, les bases de données MongoDB et Amazon DocumentDB stockent les enregistrements de données sous forme de documents regroupés dans des collections. Pour n'importe quel point de terminaison MongoDB ou Amazon DocumentDB, une base de données unique est un ensemble spécifique de collections identifiées par le nom de la base de données.

Lorsque vous migrez à partir d'une source MongoDB ou Amazon DocumentDB, vous utilisez les paramètres de chargement parallèle de manière légèrement différente. Dans ce cas, prenez en compte le type de segmentation par plage ou de segmentation automatique des paramètres de chargement parallèle pour les collections sélectionnées plutôt que les tables et les vues.

Rubriques

- Les caractères génériques dans les paramètres de table sont limités
- Utilisation du chargement parallèle pour certaines tables, vues et collections
- Spécification des paramètres LOB pour une table ou une vue sélectionnée
- Exemples de paramètres de table

Pour les règles de mappage de tables qui utilisent le type de règle des paramètres de table (table-settings), vous pouvez appliquer les paramètres suivants.

Paramètre	Valeurs possibles	Description
`rule-type`	`table-settings`	Valeur qui applique la règle à une table, une vue ou une collection spécifiée par la règle de sélection.

Paramètre	Valeurs possibles	Description
`rule-id`	Valeur numérique.	Valeur numérique unique pour identifier la règle.
`rule-name`	Valeur alphanumérique.	Nom unique pour identifier la règle.
`object-locator`	Objet avec les paramètres suivants : • `schema-name` : nom du schéma. Pour les points de terminaison MongoDB et Amazon DocumentDB, il s'agit du nom de la base de données contenant un ensemble de collections. • `table-name` : nom de la table, de la vue ou de la collection.	Le nom d'un schéma et d'une table ou d'une vue spécifiques ou le nom d'une base de données et d'une collection spécifiques (pas de caractères génériques).

Paramètre	Valeurs possibles	Description
`parallel-load`	Objet avec les paramètres suivants : • `type` : indique si le chargement parallèle est activé. Si c'est le cas, ce paramètre spécifie également le mécanisme permettant d'identifier les partition s de table ou de vue, les sous-partitions ou d'autres segments à charger en parallèle. Les partition s sont des segments qui sont déjà définis et identifiés par le nom dans la table ou la vue source. Pour les points de terminais on MongoDB et Amazon DocumentDB, les partitions sont des segments. AWS DMS peut calculer ces paramètres d'autoseg mentation automatiquement en fonction des paramètres d'autoseg mentation associés. Vous pouvez également les spécifier manuellem ent à l'aide des paramètres de segmentation par plage. Pour les points de terminaison Oracle uniquement, les sous-part itions sont un niveau supplémen taire de segments qui sont déjà définis et identifiés par le nom dans la table ou la vue source. Vous pouvez identifier d'autres segments dans la règle `table-settings` en spécifiant des limites dans	Spécifie une opération de chargement parallèle (multithread) dans la table ou la vue identifié e par l'option `object-locator` . Dans ce cas, vous pouvez charger en parallèle selon l'une de ces méthodes : • À l'aide des segments spécifiés par toutes les partitions ou sous-part itions disponibles. • À l'aide des partitions et sous-partitions sélection nées. • À l'aide de la segmentat ion automatique ou des segments basés sur une plage que vous spécifiez . Pour plus d'informations sur le chargement parallèle , consultez Utilisation du chargement parallèle pour certaines tables, vues et collections.

Paramètre	Valeurs possibles	Description
	la plage de valeurs pour une ou plusieurs colonnes de la table ou de la vue. • `partitions` : si `type` est défini sur `partitions-list`, cette valeur spécifie toutes les partitions à charger en parallèle. • `subpartitions` : pour les points de terminaison Oracle uniquement, si `type` est défini sur `partitions-list`, cette valeur spécifie toutes les sous-partitions à charger en parallèle. • `columns` : si `type` est défini sur `ranges`, cette valeur spécifie les noms des colonnes utilisés pour identifier les segments basés sur une plage à charger en parallèle. • `boundaries` : si `type` est défini sur `ranges`, cette valeur spécifie les valeurs de `columns` utilisées pour identifier les segments basés sur une plage à charger en parallèle.	

Paramètre	Valeurs possibles	Description
`type`	L'une des options suivantes pour `parallel-load` : • `partitions-auto` : toutes les partitions de la table ou de la vue sont chargées en parallèle. Chaque partition est allouée à son propre thread. Ce paramètre est obligatoire pour que les points de terminaison sources MongoDB et Amazon DocumentDB puissent utiliser l'option de segmentation automatique d'un chargement complet parallèle. • `subpartitions-auto` : (points de terminaison Oracle uniquement) Toutes les sous-partitions de la table ou de la vue sont chargées en parallèle. Chaque sous-partition est allouée à son propre thread. • `partitions-list` : toutes les partitions spécifiées de la table ou de la vue sont chargées en parallèle. Pour les points de terminaison Oracle uniquement, toutes les sous-partitions spécifiées de la table ou de la vue sont chargées en parallèle. Chaque partition et sous-partition que vous spécifiez sont allouées à son propre thread. Vous identifiez les partitions et les sous-partitions à	Mécanisme permettant d'identifier les partitions, les sous-partitions ou les segments de table, de vue ou de collection à charger en parallèle.

Paramètre	Valeurs possibles	Description
	charger en parallèle par noms de partition (`partitions`) et noms de sous-partition (`subpartitions`). • `ranges` : tous les segments spécifiés par plage de la table, de la vue ou de la collection sont chargés en parallèle. Chaque segment de table, de vue ou de collection que vous identifiez est alloué à son propre thread. Vous spécifiez ces segments par noms de colonne (`columns`) et valeurs de colonne (`boundaries`). Les points de terminaison PostgreSQL ne prennent en charge que ce type de chargement parallèle. MongoDB et Amazon DocumentDB en tant que points de terminaison sources prennent en charge à la fois ce type de segmentation par plage et le type de segmentation automatique d'un chargement complet parallèle (`partitions-auto`). • `none` : la table, la vue ou la collection est chargée dans une tâche à thread unique (valeur par défaut), quelles que soient ses partitions ou sous-partitions. Pour plus d'informations, consultez Création d'une tâche.	

Paramètre	Valeurs possibles	Description
`number-of-partitions`	(Facultatif) Lorsque `type` est défini sur `partitions-auto` pour les collections spécifiées d'un point de terminaison MongoDB ou Amazon DocumentDB, ce paramètre indique le nombre total de partitions (segments) utilisées pour la migration . La valeur par défaut est 16.	Spécifie le nombre exact de partitions à charger en parallèle.
`collection-count-from-metadata`	(Facultatif) Lorsqu'il `type` s'agit `partitions-auto` de collections spécifiées d'un point de terminaison MongoDB ou Amazon DocumentDB et que ce paramètre est défini sur `true` AWS DMS , utilise un nombre de collections estimé pour déterminer le nombre de partitions. Si ce paramètre est défini sur`false`, AWS DMS utilise le nombre réel de collectes. L'argument par défaut est `true`.	Indique s'il convient d'utiliser une estimation du nombre de collections ou le nombre réel de collections pour calculer le nombre de partitions à charger en parallèle.

Paramètre	Valeurs possibles	Description
`max-records-skip-per-page`	(Facultatif) Lorsque `type` est défini sur `partitions-auto` pour les collections spécifiées d'un point de terminaison MongoDB ou Amazon DocumentDB, il s'agit du nombre d'enregistrements à ignorer simultanément lors de la détermination des limites pour chaque partition. AWS DMS ignore un certain nombre d'enregistrements par page pour déterminer la limite minimale d'une partition. La valeur par défaut est 10 000.	Spécifie le nombre d'enregistrements à ignorer simultanément lors de la détermination des limites de chaque partition. La définition d'une valeur relativement élevée à partir de la valeur par défaut peut entraîner des délais d'expiration du curseur et des échecs de tâche. La définition d'une valeur relativement faible à partir de la valeur par défaut entraîne un plus grand nombre d'opérations par page et ralentit le chargement complet.
`batch-size`	(Facultatif) Lorsque `type` est défini sur `partitions-auto` pour les collections spécifiées d'un point de terminaison MongoDB ou Amazon DocumentDB, cette valeur entière limite le nombre de documents renvoyés par lot aller-retour. Si la taille du lot est égale à zéro (0), le curseur utilise la taille de lot maximale définie par le serveur. La valeur par défaut est 0.	Spécifie le nombre maximal de documents renvoyés par lot. Chaque lot nécessite un aller retour jusqu'au serveur.
`partitions`	Lorsque `type` est `partitions-list` , il s'agit d'un tableau de chaînes qui spécifient les noms des partitions à charger en parallèle.	Noms des partitions à charger en parallèle.

Paramètre	Valeurs possibles	Description
`subpartitions`	(Points de terminaison Oracle uniquement) Lorsque `type` est `partitions-list` , il s'agit d'un tableau de chaînes qui spécifient les noms des sous-partitions à charger en parallèle.	Noms des sous-partitions à charger en parallèle.
`columns`	Lorsque `type` est défini sur `ranges`, il s'agit d'un tableau de chaînes définies sur les noms des colonnes utilisées pour identifier les segments de table, de vue ou de collection basés sur une plage à charger en parallèle.	Noms des colonnes utilisés pour identifier les segments de table, de vue ou de collection basés sur une plage à charger en parallèle.
`boundaries`	Lorsque `type` est `ranges`, tableau de tableaux de valeurs de colonne. Chaque tableau de valeurs de colonne contient des valeurs de colonne dans la quantité et l'ordre spécifiés par `columns`. Un tableau de valeurs de colonne spécifie la limite supérieure d'un segment de table, de vue ou de collectio n. Chaque tableau de valeurs de colonne supplémentaire ajoute la limite supérieure pour un segment de table, de vue ou de collection supplémentaire. Tous ces segments de table, de vue ou de collection basés sur une plage sont chargés en parallèle.	Valeurs de colonne qui identifient les partitions de table, de vue ou de collection basées sur une plage à charger en parallèle.

Paramètre	Valeurs possibles	Description
`lob-settings`	Objet avec les paramètres suivants : • `mode` : spécifie le mode de gestion de migration des objets LOB. • `bulk-max-size` : spécifie la taille maximale des objets LOB, en fonction du paramètre `mode`.	Valeur qui spécifie la gestion LOB pour la table ou la vue identifié e par l'option `object-locator` . La tâche de gestion LOB spécifiée remplace tous les paramètres de tâche LOB de cette table ou de cette vue uniquement. Pour plus d'informations sur l'utilisa tion des paramètres LOB, consultez Spécification des paramètres LOB pour une table ou une vue sélection née.

Paramètre	Valeurs possibles	Description
`mode`	Spécifie le traitement de migration des LOB dans la table ou la vue sélectionnée à l'aide des valeurs suivantes : • `limited` : (par défaut) Cette valeur définit la migration en mode LOB limité, avec tous les objets LOB migrés en ligne, ainsi que tous les autres types de données des colonnes de la table ou de la vue. Utilisez cette valeur lors de la réplication des LOB de petite taille (100 Mo ou moins). Spécifiez également une valeur `bulk-max-size` (zéro n'est pas valide). Tous les LOB ayant migré supérieurs à `bulk-max-size` sont tronqués à la taille que vous définissez. • `unlimited` : cette valeur définit la migration en mode LOB complet. Utilisez cette valeur lorsque la totalité ou la majorité des LOB que vous souhaitez répliquer ont une taille supérieure à 1 Go. Si vous spécifiez une valeur `bulk-max-size` égale à zéro, tous les LOB sont migrés en mode LOB complet standard. Dans cette forme de mode `unlimited` , tous les LOB sont migrés séparément à partir des autres types de données de colonne à l'aide d'une recherche dans la table ou la vue source. Si	Spécifie le mécanisme utilisé pour migrer les LOB.

Paramètre	Valeurs possibles	Description
	vous spécifiez une valeur `bulk-max-size` supérieure à zéro, tous les LOB sont migrés en mode LOB complet combinaison. Dans cette forme de mode `unlimited`, les LOB supérieurs à `bulk-max-size` sont migrés à l'aide d'une recherche dans la table ou la vue source, similaire au mode LOB complet standard. Sinon, les LOB inférieurs ou égaux à cette taille sont migrés en ligne, comme dans le mode LOB limité. Aucun LOB n'est jamais tronqué en mode `unlimited`, quelle que soit la forme que vous utilisez. • `none` : tous les objets LOB de table ou de vue sont migrés selon les paramètres LOB de tâche. Pour plus d'informations sur les paramètres LOB de tâche, consultez Paramètres de métadonnées des tâches cibles. Pour plus d'informations sur la migration des objets LOB et sur la spécification de ces paramètres LOB de tâche, consultez Configuration du support LOB pour les bases de données sources dans une tâche AWS DMS.	

Paramètre	Valeurs possibles	Description
`bulk-max-size`	L'effet de cette valeur dépend du `mode`.	Spécifie la taille maximale des LOB en kilo-octets par incréments. Spécifiez cette option uniquement si vous avez besoin de répliquer des LOB de petite taille ou si le point de terminaison cible ne prend pas en charge la taille de LOB illimitée.

Les caractères génériques dans les paramètres de table sont limités

L'utilisation du caractère générique de pourcentage (`"%"`) dans les règles `"table-settings"` n'est pas prise en charge pour les bases de données sources, comme indiqué ci-dessous.

```
{
    "rule-type": "table-settings",
    "rule-id": "8",
    "rule-name": "8",
    "object-locator": {
        "schema-name": "ipipeline-prod",
        "table-name": "%"
    },
    "parallel-load": {
        "type": "partitions-auto",
        "number-of-partitions": 16,
        "collection-count-from-metadata": "true",
        "max-records-skip-per-page": 1000000,
        "batch-size": 50000
    }
  }
```

Si vous utilisez `"%"` les `"table-settings"` règles comme indiqué, AWS DMS renvoie alors l'exception suivante.

```
Error in mapping rules. Rule with ruleId = x failed validation. Exact
```

```
schema and table name required when using table settings rule.
```

En outre, il vous AWS recommande de ne pas charger un grand nombre de collections volumineuses en une seule tâche avec`parallel-load`. Notez que AWS DMS limite la contention des ressources ainsi que le nombre de segments chargés en parallèle par la valeur du paramètre de tâche `MaxFullLoadSubTasks`, avec une valeur maximale de 49.

Spécifiez plutôt toutes les collections de la base de données source pour les plus grandes collections en spécifiant individuellement chaque `"schema-name"` et `"table-name"`. Augmentez également votre migration de manière appropriée. Par exemple, exécutez plusieurs tâches sur un nombre suffisant d'instances de réplication pour gérer un grand nombre de collections volumineuses dans la base de données.

Utilisation du chargement parallèle pour certaines tables, vues et collections

Pour accélérer la migration et la rendre plus efficace, vous pouvez utiliser le chargement parallèle pour certaines tables, vues et collections relationnelles. En d'autres termes, vous pouvez migrer une seule table, vue ou collection segmentée en utilisant plusieurs threads en parallèle. Pour ce faire, AWS DMS divisez une tâche à chargement complet en fils, chaque segment de table étant affecté à son propre fil.

À l'aide de ce processus de chargement parallèle, plusieurs threads peuvent décharger plusieurs tables, vues et collections en parallèle à partir du point de terminaison source. Ensuite, plusieurs threads peuvent migrer et charger les mêmes tables, vues et collections en parallèle vers le point de terminaison cible. Pour certains moteurs de base de données, vous pouvez segmenter les tables et vues par partitions ou sous-partitions existantes. Pour les autres moteurs de base de données, vous pouvez segmenter AWS DMS automatiquement les collections en fonction de paramètres spécifiques (autosegmentation). Sinon, vous pouvez segmenter n'importe quelle table, vue ou collection par des plages de valeurs de colonne que vous spécifiez.

Le chargement parallèle est pris en charge pour les points de terminaison source suivants :

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2
- SAP Adaptive Server Enterprise (ASE)

- MongoDB (prend uniquement en charge les options de segmentation automatique et de segmentation par plage d'un chargement complet parallèle)
- Amazon DocumentDB (prend uniquement en charge les options de segmentation automatique et de segmentation par plage d'un chargement complet parallèle)
- Amazon DynamoDB

Pour les points de terminaison MongoDB et Amazon DocumentDB AWS DMS , prend en charge les types de données suivants pour les colonnes qui sont des clés de partition pour l'option de segmentation de plage d'un chargement complet en parallèle.

- Double
- Chaîne
- ObjectId
- Entier 32 bits
- Entier 64 bits

La charge parallèle à utiliser avec les règles de définition de table est prise en charge pour les points de terminaison cibles suivants :

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- Amazon S3
- SAP Adaptive Server Enterprise (ASE)
- Amazon Redshift
- MongoDB (prend uniquement en charge les options de segmentation automatique et de segmentation par plage d'un chargement complet parallèle)
- Amazon DocumentDB (prend uniquement en charge les options de segmentation automatique et de segmentation par plage d'un chargement complet parallèle)

Pour spécifier le nombre maximal de tables et de vues à charger en parallèle, utilisez le paramètre de tâche `MaxFullLoadSubTasks`.

Pour spécifier le nombre maximal de threads par table ou vue pour les cibles prises en charge d'une tâche de chargement parallèle, définissez davantage de segments à l'aide de limites de valeurs de colonnes.

> **Important**
>
> `MaxFullLoadSubTasks` contrôle le nombre de tables ou de segments de table à charger en parallèle. `ParallelLoadThreads` contrôle le nombre de threads utilisés par une tâche de migration pour exécuter les chargements en parallèle. Ces paramètres sont multiplicatifs. Ainsi, le nombre total de threads utilisés lors d'une tâche de chargement complet correspond approximativement au résultat de la valeur de `ParallelLoadThreads` multipliée par la valeur de `MaxFullLoadSubTasks` (`ParallelLoadThreads` * `MaxFullLoadSubtasks`)). Si vous créez des tâches avec un grand nombre de sous-tâches de chargement complet et un grand nombre de threads de chargement parallèle, votre tâche peut consommer trop de mémoire et échouer.

Pour spécifier le nombre maximal de threads par table pour les cibles Amazon DynamoDB, Amazon Kinesis Data Streams, Apache Kafka ou Amazon Elasticsearch Service, utilisez le paramètre de tâche de métadonnées cible `ParallelLoadThreads`.

Pour spécifier la taille du tampon pour une tâche de chargement parallèle lorsque `ParallelLoadThreads` est utilisé, utilisez le paramètre de tâche de métadonnées cible `ParallelLoadBufferSize`.

La disponibilité et les paramètres de `ParallelLoadThreads` et `ParallelLoadBufferSize` dépendent du point de terminaison cible.

Pour plus d'informations sur les paramètres `ParallelLoadThreads` et `ParallelLoadBufferSize`, consultez Paramètres de métadonnées des tâches cibles. Pour plus d'informations sur le paramètre `MaxFullLoadSubTasks`, consultez Paramètres de tâche de chargement complet. Pour obtenir des informations spécifiques sur les points de terminaison cible, consultez les rubriques connexes.

Pour utiliser le chargement parallèle, créez une règle de mappage de table de type `table-settings` avec l'option `parallel-load`. Dans la règle `table-settings`, vous pouvez spécifier les critères de segmentation d'une table, d'une vue ou d'une collection unique que vous voulez charger en parallèle. Pour ce faire, définissez le paramètre `type` de l'option `parallel-load` sur l'une de plusieurs options.

Le choix effectué dépendra de la façon dont vous souhaitez segmenter la table, la vue ou la collection pour le chargement parallèle :

- Par partitions (ou segments) : chargez toutes les partitions (ou segments) de table ou de vue existantes à l'aide du type `partitions-auto`. Sinon, chargez uniquement les partitions sélectionnées à l'aide du type `partitions-list` avec un tableau de partitions spécifié.

  Pour les points de terminaison MongoDB et Amazon DocumentDB uniquement, chargez toutes les collections ou les collections spécifiées par segments AWS DMS qui calculent automatiquement également en utilisant `partitions-auto` le type et des paramètres facultatifs supplémentaires. `table-settings`
- (Points de terminaison Oracle uniquement) Par sous-partitions : chargez toutes les sous-partitions de table ou de vue existantes à l'aide du type `subpartitions-auto`. Ou chargez uniquement les sous-partitions sélectionnées en utilisant le type `partitions-list` avec un tableau `subpartitions` spécifié.
- Par segments que vous définissez : chargez les segments de table, de vue ou de collection que vous définissez à l'aide des limites de valeur de colonne. Pour ce faire, utilisez le type `ranges` avec les tableaux `columns` et `boundaries` spécifiés.

> **Note**
>
> Les points de terminaison PostgreSQL ne prennent en charge que ce type de chargement parallèle. MongoDB et Amazon DocumentDB en tant que points de terminaison sources prennent en charge à la fois ce type de segmentation par plage et le type de segmentation automatique d'un chargement complet parallèle (`partitions-auto`).

Pour identifier les tables, les vues ou les collections supplémentaires à charger en parallèle, spécifiez des objets `table-settings` supplémentaires avec les options `parallel-load`.

Les procédures suivantes décrivent comment coder le JSON pour chaque type de chargement parallèle, du plus simple au plus complexe.

Pour spécifier toutes les partitions de table, de vue ou de collection, ou toutes les sous-partitions de table ou de vue

- Spécifiez `parallel-load` avec le type `partitions-auto` ou le type `subpartitions-auto` (mais pas les deux).

Chaque partition (ou segment) ou sous-partition de table, de vue ou de collection est alors allouée automatiquement à son propre thread.

Pour certains points de terminaison, le chargement parallèle comporte des partitions ou sous-partitions uniquement si elles sont déjà définies pour la table ou la vue. Pour les points de terminaison sources MongoDB et Amazon DocumentDB, vous pouvez calculer automatiquement AWS DMS les partitions (ou les segments) en fonction de paramètres supplémentaires facultatifs. Cela inclut `number-of-partitions`, `collection-count-from-metadata`, `max-records-skip-per-page` et `batch-size`.

Pour spécifier des partitions ou des sous-partitions de table ou de vue sélectionnées, ou les deux

1. Spécifiez `parallel-load` avec le type `partitions-list`.
2. (Facultatif) Incluez les partitions en spécifiant une liste de noms de partitions comme valeur de `partitions`.

   Chaque partition spécifiée est alors allouée à son propre thread.

   > **Important**
   >
   > Pour les points de terminaison Oracle, assurez-vous que les partitions et les sous-partitions ne se chevauchent pas lorsque vous les choisissez pour le chargement parallèle. Si vous utilisez des partitions et des sous-partitions qui se chevauchent pour charger des données en parallèle, les entrées sont dupliquées ou le chargement échoue en raison d'une violation de clé primaire en double.

3. (Facultatif) Pour les points de terminaison Oracle uniquement, incluez les sous-partitions en définissant un tableau de noms de sous-partitions comme valeur de `subpartitions`.

   Chaque sous-partition spécifiée est alors allouée à son propre thread.

   > **Note**
   >
   > Le chargement parallèle comporte des partitions ou sous-partitions uniquement si elles sont déjà définies pour la table ou la vue.

Vous pouvez spécifier des segments de table ou de vue en tant que plages de valeurs de colonne. Lorsque vous procédez ainsi, vous devez être conscient de ces caractéristiques de la colonne :

- La spécification de colonnes indexées améliore considérablement les performances.
- Vous pouvez spécifier jusqu'à 10 colonnes.
- Vous ne pouvez pas utiliser de colonnes pour définir les limites des segments avec les types de AWS DMS données suivants : DOUBLE, FLOAT, BLOB, CLOB et NCLOB
- Les enregistrements avec des valeurs null ne sont pas répliqués.

Pour spécifier des segments de table, de vue ou de collection en tant que plages de valeurs de colonne

1. Spécifiez `parallel-load` avec le type `ranges`.
2. Définissez une limite entre les segments de table ou de vue en spécifiant une liste de noms de colonne comme valeur de `columns`. Faites la même chose pour chaque colonne pour laquelle vous souhaitez définir une limite entre les segments de table ou de vue.

   Notez que l'ordre des colonnes est significatif. Ici, la première colonne est la plus significative et la dernière colonne est la moins significative pour la définition de chaque limite, comme décrit ci-après.
3. Pour définir les plages de données pour tous les segments de table ou de vue, spécifiez un tableau de limites comme valeur pour `boundaries`. Un tableau de limites est un tableau de tableaux de valeurs de colonne. Pour ce faire, effectuez les étapes suivantes :

   a. Spécifiez chaque élément d'un tableau de valeurs de colonne comme valeur qui correspond à chaque colonne. Un tableau de valeurs de colonne représente la limite supérieure de chaque segment de table ou de vue que vous souhaitez définir. Spécifiez chaque colonne dans le même ordre que celui où vous avez spécifié cette colonne dans le tableau `columns`.

      Entrez les valeurs des colonnes DATE au format pris en charge par la source.

   b. Spécifiez chaque tableau de valeurs de colonne comme limite supérieure, dans l'ordre, de chaque segment, du bas au next-to-top segment de la table ou de la vue. S'il existe des lignes au-dessus de la limite supérieure que vous spécifiez, ces lignes complètent le segment supérieur de la table ou de la vue. Par conséquent, le nombre de segments basés sur une plage est potentiellement supérieur au nombre de limites de segment dans le

tableau de limites. Chaque segment basé sur une plage de ce type est alloué à son propre thread.

Toutes les données non null sont répliquées, même si vous ne définissez pas de plages de données pour toutes les colonnes de la table ou de la vue.

Par exemple, supposons que vous définissez trois tableaux de valeurs de colonne pour les colonnes COL1, COL2 et COL3, comme suit.

COL1	COL2	COL3
10	30	105
20	20	120
100	12	99

Vous avez défini trois limites de segment pour un total possible de quatre segments.

Pour identifier les plages de lignes à répliquer pour chaque segment, l'instance de réplication applique une recherche à ces trois colonnes pour chacun des quatre segments : La recherche est similaire à ce qui suit :

Segment 1

Répliquez toutes les lignes où les conditions suivantes sont réunies : les valeurs des deux premières colonnes sont inférieures ou égales aux valeurs de limite supérieure de segment 1 correspondantes. En outre, les valeurs de la troisième colonne sont inférieures aux valeurs de limite supérieure de segment 1.

Segment 2

Répliquez toutes les lignes (sauf les lignes segment 1) où les conditions suivantes sont réunies : les valeurs des deux premières colonnes sont inférieures ou égales aux valeurs de limite supérieure de segment 2 correspondantes. En outre, les valeurs de la troisième colonne sont inférieures aux valeurs de limite supérieure de segment 2.

Segment 3

Répliquez toutes les lignes (sauf les lignes segment 2) où les conditions suivantes sont réunies : les valeurs des deux premières colonnes sont inférieures ou égales aux valeurs de limite supérieure de segment 3 correspondantes. En outre, les valeurs de la troisième colonne sont inférieures aux valeurs de limite supérieure de segment 3 .

Segment 4

Répliquer toutes les lignes restantes (sauf les lignes de Segment 1, 2 et 3).

Dans ce cas, l'instance de réplication crée une clause `WHERE` pour charger chaque segment comme suit :

Segment 1

```
((COL1 < 10) OR ((COL1 = 10) AND (COL2 < 30)) OR ((COL1 = 10) AND
(COL2 = 30) AND (COL3 < 105)))
```

Segment 2

```
NOT ((COL1 < 10) OR ((COL1 = 10) AND (COL2 < 30)) OR ((COL1 = 10) AND
(COL2 = 30) AND (COL3 < 105))) AND ((COL1 < 20) OR ((COL1 = 20) AND
(COL2 < 20)) OR ((COL1 = 20) AND (COL2 = 20) AND (COL3 < 120)))
```

Segment 3

```
NOT ((COL1 < 20) OR ((COL1 = 20) AND (COL2 < 20)) OR ((COL1 = 20) AND
(COL2 = 20) AND (COL3 < 120))) AND ((COL1 < 100) OR ((COL1 = 100) AND
(COL2 < 12)) OR ((COL1 = 100) AND (COL2 = 12) AND (COL3 < 99)))
```

Segment 4

```
NOT ((COL1 < 100) OR ((COL1 = 100) AND (COL2 < 12)) OR ((COL1 = 100)
AND (COL2 = 12) AND (COL3 < 99)))
```

Spécification des paramètres LOB pour une table ou une vue sélectionnée

Vous pouvez définir des paramètres LOB de tâche pour une ou plusieurs tables en créant une règle de mappage de table de type `table-settings` avec l'option `lob-settings` pour un ou plusieurs objets `table-settings`.

La spécification des paramètres LOB pour les tables ou vues sélectionnées est prise en charge pour les points de terminaison source suivants :

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2, selon les paramètres `mode` et `bulk-max-size`, décrits ci-après
- SAP Adaptive Server Enterprise (ASE), selon les paramètres `mode` et `bulk-max-size`, comme décrit ci-après.

La spécification des paramètres LOB pour les tables ou vues sélectionnées est prise en charge pour les points de terminaison cible suivants :

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- SAP ASE, selon les paramètres `mode` et `bulk-max-size`, comme décrit ci-après.

> **Note**
>
> Vous ne pouvez utiliser les types de données LOB qu'avec les tables et les vues qui incluent une clé primaire.

Pour utiliser des paramètres LOB pour une table ou une vue sélectionnée, créez une règle de mappage de table de type `table-settings` avec l'option `lob-settings`. Cette option indique le traitement des LOB pour la table ou la vue identifiée par l'option `object-locator`. Dans la règle `table-settings`, vous pouvez spécifier un objet `lob-settings` avec les paramètres suivants :

- `mode` : spécifie le mécanisme de traitement de la migration des objets LOB pour la table ou la vue sélectionnée, comme suit :
  - `limited` : le mode LOB limité par défaut est le plus rapide et le plus efficace. Utilisez ce mode uniquement si tous vos LOB sont petits (moins de 100 Mo) ou si le point de terminaison cible ne

prend pas en charge une taille de LOB illimitée. De plus, si vous utilisez `limited`, tous les LOB doivent se trouver dans la limite de taille que vous avez définie pour `bulk-max-size`.

Dans ce mode, pour une tâche de chargement complet, l'instance de réplication migre tous les LOB en ligne avec les autres types de données de colonne dans le cadre d'un stockage de table ou de vue principal. Toutefois, l'instance tronque les LOB supérieurs à votre valeur `bulk-max-size` à la taille spécifiée. Pour un chargement de capture des données modifiées (CDC), l'instance migre tous les LOB à l'aide d'une recherche de table source, quelle que soit la taille des LOB, comme dans le mode LOB complet standard. Il procède ainsi quelle que soit la taille du LOB.

> **Note**
>
> Vous ne pouvez migrer des vues que pour les tâches de chargement complet.

- `unlimited` : le mécanisme de migration pour le mode LOB complet dépend de la valeur que vous définissez pour `bulk-max-size`, comme suit :
  - Mode LOB complet standard : lorsque vous définissez `bulk-max-size` sur zéro, l'instance de réplication migre tous les objets LOB à l'aide du mode LOB complet standard. Ce mode nécessite une recherche dans la table ou la vue source afin de migrer tous les LOB, quelle que soit leur taille. Cela entraîne généralement une migration un peu plus lente que pour le mode LOB limité. Utilisez ce mode uniquement si la totalité ou la plupart de vos LOB sont volumineux (1 Go ou plus).
  - Mode LOB complet avec combinaison : lorsque vous définissez `bulk-max-size` sur une valeur différente de zéro, ce mode LOB complet utilise une combinaison du mode LOB limité et du mode LOB complet standard. En d'autres termes, si une taille de LOB est incluse dans votre valeur `bulk-max-size`, l'instance migre le LOB en ligne comme dans le mode LOB limité. Si la taille de LOB est supérieure à cette valeur, l'instance migre le LOB à l'aide d'une recherche dans la table ou la vue source comme dans le mode LOB complet standard. Pour un chargement de capture des données modifiées (CDC), l'instance migre tous les LOB à l'aide d'une recherche de table source, quelle que soit la taille des LOB, comme dans le mode LOB complet standard. Il procède ainsi quelle que soit la taille du LOB.

    > **Note**
    >
    > Vous ne pouvez migrer des vues que pour les tâches de chargement complet.

Ce mode entraîne une vitesse de migration constituant un compromis entre le mode LOB limité plus rapide et le mode LOB complet standard plus lent. Utilisez ce mode uniquement lorsque vous avez une combinaison de LOB de petites et grandes tailles et que la plupart des LOB sont petits.

Ce mode LOB complet avec combinaison n'est disponible que pour les points de terminaison suivants :

- IBM Db2 en tant que source
- SAP ASE comme source ou cible

Quelle que soit la façon dont vous spécifiez le mode `unlimited`, l'instance migre tous les LOB entièrement, sans troncature.

- `none` : l'instance de réplication migre les objets LOB de la table ou la vue sélectionnée à l'aide de vos paramètres LOB de tâche. Vous pouvez utiliser cette option pour comparer les résultats de la migration avec et sans paramètres LOB pour la table ou la vue sélectionnée.

Si la table ou la vue spécifiée a des LOB inclus dans la réplication, vous ne pouvez définir le paramètre de tâche `BatchApplyEnabled` sur `true` qu'en cas d'utilisation du mode LOB `limited`.

Dans certains cas, il se peut que vous définissiez `BatchApplyEnabled` sur `true` et `BatchApplyPreserveTransaction` sur `false`. Dans ce cas, l'instance définit `BatchApplyPreserveTransaction` sur `true` si la table ou la vue a des LOB et que les points de terminaison source et cible sont Oracle.

- `bulk-max-size` : définissez cette valeur sur zéro ou sur une valeur différente de zéro en kilo-octets, selon le `mode`, comme décrit dans les éléments précédents. En mode `limited`, vous devez définir une valeur différente de zéro pour ce paramètre.

  L'instance convertit les LOB au format binaire. Par conséquent, pour spécifier le LOB de plus grande taille que vous devez répliquer, multipliez sa taille par trois. Par exemple, si la taille du LOB le plus grand est de 2 Mo, définissez `bulk-max-size` sur 6 000 (6 Mo).

Exemples de paramètres de table

Vous trouverez ci-dessous des exemples qui illustrent l'utilisation des paramètres de table.

Example Charger une table segmentée par partitions

L'exemple suivant charge une table SALES de votre source plus efficacement en la chargeant en parallèle sur la base de toutes ses partitions.

```
{
   "rules": [{
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "HR",
                "table-name": "SALES"
            },
            "parallel-load": {
                "type": "partitions-auto"
            }
        }
     ]
}
```

Example Charger une table segmentée par sous-partitions

L'exemple suivant charge une table SALES dans votre source Oracle plus efficacement en la chargeant en parallèle sur la base de toutes ses sous-partitions.

```
{
   "rules": [{
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
```

```
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "HR",
                "table-name": "SALES"
            },
            "parallel-load": {
                "type": "subpartitions-auto"
            }
        }
     ]
}
```

Example Charger une table segmentée par une liste de partitions

L'exemple suivant charge une table SALES de votre source en la chargeant en parallèle sur la base d'une liste de partitions particulière. Ici, les partitions spécifiées sont nommées à partir de valeurs commençant par des parties de l'alphabet, par exemple, ABCD, EFGH, etc.

```
{
    "rules": [{
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "HR",
                "table-name": "SALES"
```

```
            },
            "parallel-load": {
                "type": "partitions-list",
                "partitions": [
                    "ABCD",
                    "EFGH",
                    "IJKL",
                    "MNOP",
                    "QRST",
                    "UVWXYZ"
                ]
            }
        }
    ]
}
```

Example Charger une table Oracle segmentée par une liste sélectionnée de partitions et sous-partitions

L'exemple suivant charge une table SALES de votre source Oracle en la chargeant en parallèle sur la base d'une liste sélectionnée de partitions et sous-partitions. Ici, les partitions spécifiées sont nommées à partir de valeurs commençant par des parties de l'alphabet, par exemple, ABCD, EFGH, etc. Les sous-partitions spécifiées sont nommés à partir de valeurs commençant par des chiffres, par exemple, 01234 et 56789.

```
{
    "rules": [{
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "HR",
                "table-name": "SALES"
```

```
            },
            "parallel-load": {
                "type": "partitions-list",
                "partitions": [
                    "ABCD",
                    "EFGH",
                    "IJKL",
                    "MNOP",
                    "QRST",
                    "UVWXYZ"
                ],
                "subpartitions": [
                    "01234",
                    "56789"
                ]
            }
        }
    ]
}
```

Example Charger une table segmentée par des plages de valeurs de colonne

L'exemple suivant charge une table SALES de votre source en la chargeant en parallèle par les segments spécifiés par les plages de valeurs de colonne SALES_NO et REGION.

```
{
    "rules": [{
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "HR",
                "table-name": "SALES"
```

```
            },
            "parallel-load": {
                "type": "ranges",
                "columns": [
                    "SALES_NO",
                    "REGION"
                ],
                "boundaries": [
                    [
                        "1000",
                        "NORTH"
                    ],
                    [
                        "3000",
                        "WEST"
                    ]
                ]
            }
        }
    ]
}
```

Ici, deux colonnes sont spécifiées pour les plages de segment avec les noms `SALES_NO` et `REGION`. Deux limites sont spécifiées avec deux ensembles de valeurs de colonne (`["1000","NORTH"]` et `["3000","WEST"]`).

Ces deux limites identifient donc les trois segments de table suivants à charger en parallèle :

Segment 1

Lignes avec des valeurs `SALES_NO` inférieures ou égales à 1 000 et `REGION` inférieures à « NORTH ». En d'autres termes, les chiffres de vente jusqu'à 1 000 dans la région EAST.

Segment 2

Lignes autres que Segment 1 avec des valeurs `SALES_NO` inférieures ou égales à 3 000 et `REGION` inférieures à « WEST ». En d'autres termes, les chiffres de vente compris entre 1 000 et 3 000 dans les régions NORTH et SOUTH.

Segment 3

Toutes les lignes restantes autres que Segment 1 et Segment 2. En d'autres termes, les chiffres de vente supérieurs à 3 000 dans la région WEST.

Example Charger deux tables : une segmentée par plages et une autre segmentée par partitions

L'exemple suivant charge une table SALES en parallèle par les limites de segment que vous identifiez. Il charge également une table ORDERS en parallèle par l'ensemble de ses partitions, comme avec les exemples précédents.

```
{
    "rules": [{
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "HR",
                "table-name": "SALES"
            },
            "parallel-load": {
                "type": "ranges",
                "columns": [
                    "SALES_NO",
                    "REGION"
                ],
                "boundaries": [
                    [
                        "1000",
                        "NORTH"
                    ],
                    [
                        "3000",
                        "WEST"
                    ]
                ]
            }
        },
```

```
        {
            "rule-type": "table-settings",
            "rule-id": "3",
            "rule-name": "3",
            "object-locator": {
                "schema-name": "HR",
                "table-name": "ORDERS"
            },
            "parallel-load": {
                "type": "partitions-auto"
            }
        }
    ]
}
```

Example Charger une table avec des LOB à l'aide du mode LOB limité

L'exemple suivant charge une table `ITEMS` incluant des LOB de votre source à l'aide du mode LOB limité (valeur par défaut) avec une taille maximale non tronquée de 100 Mo. Les LOB qui dépassent cette taille sont tronqués à 100 Mo. Tous les LOB sont chargés en ligne avec tous les autres types de données de colonne.

```
{
   "rules": [{
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "INV",
                "table-name": "ITEMS"
            },
            "lob-settings": {
                "bulk-max-size": "100000"
```

```
            }
        }
    ]
}
```

Example Charger une table avec des LOB à l'aide du mode LOB complet standard

L'exemple suivant charge une table `ITEMS` de votre source, y compris tous ses LOB sans troncation à l'aide du mode LOB complet standard. Tous les LOB, quelle que soit leur taille, sont chargés séparément des autres types de données à l'aide d'une recherche de chaque LOB dans la table source.

```
{
   "rules": [{
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "INV",
                "table-name": "ITEMS"
            },
            "lob-settings": {
                "mode": "unlimited",
                "bulk-max-size": "0"
            }
        }
    ]
}
```

Example Charger une table avec des LOB à l'aide du mode LOB complet avec combinaison

L'exemple suivant charge une table `ITEMS` de votre source, y compris tous ses LOB sans troncation à l'aide du mode LOB complet avec combinaison Tous les LOB dont la taille est inférieure à 100 Mo sont chargés en ligne avec les autres types de données, comme dans le mode LOB limité. Tous les LOB supérieurs à 100 Mo sont chargés séparément des autres types de données. Ce chargement distinct utilise une recherche pour chaque LOB dans la table source, comme dans le mode LOB complet standard.

```
{
   "rules": [{
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "INV",
                "table-name": "ITEMS"
            },
            "lob-settings": {
                "mode": "unlimited",
                "bulk-max-size": "100000"
            }
        }
     ]
}
```

Example Charger une table avec des LOB à l'aide des paramètres LOB de tâche

L'exemple suivant charge une table `ITEMS` de votre source, y compris tous les LOB, à l'aide de ses paramètres LOB de tâche. Le paramètre `bulk-max-size` avec la valeur 100 Mo est ignoré et conservé uniquement pour une réinitialisation rapide au mode `limited` ou `unlimited`.

```
{
   "rules": [{
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "INV",
                "table-name": "ITEMS"
            },
            "lob-settings": {
                "mode": "none",
                "bulk-max-size": "100000"
            }
        }
     ]
}
```

Utilisation de filtres de source

Vous pouvez utiliser des filtres de source pour limiter le nombre et le type d'enregistrements transférés de votre source vers votre cible. Par exemple, vous pouvez spécifier que seuls les employés avec un emplacement de siège sont déplacés vers la base de données cible. Les filtres font partie d'une règle de sélection. Vous appliquez des filtres sur une colonne de données.

Les filtres de source doivent respecter les contraintes suivantes :

- Une règle de sélection peut avoir aucun filtre ou un ou plusieurs filtres.
- Chaque filtre peut avoir une ou plusieurs conditions de filtre.
- Si plusieurs filtres sont utilisés, la liste des filtres est combinée comme si elle utilisait l'opérateur AND entre les filtres.

- Si plusieurs conditions de filtre sont utilisées au sein d'un seul filtre, la liste des conditions de filtre est combinée comme si elle utilisait un opérateur OR entre les conditions de filtre.
- Les filtres sont appliqués uniquement quand `rule-action = 'include'`.
- Filtres nécessitent un nom de colonne et une liste de conditions de filtre. Les conditions de filtre doivent comporter un opérateur de filtre associé à une valeur, deux valeurs ou aucune valeur, selon l'opérateur.
- Les noms de colonnes, de tables, de vues et de schémas sont sensibles à la casse. Oracle et Db2 doivent toujours utiliser des MAJUSCULES.
- Les filtres ne prennent en charge que les tables portant des noms exacts. Les filtres ne prennent pas en charge les caractères génériques.

Les limitations suivantes s'appliquent à l'utilisation des filtres de source :

- Les filtres ne calculent pas les colonnes de langues écrites de droite à gauche.
- N'appliquez pas de filtre aux colonnes LOB.
- Appliquez des filtres uniquement aux colonnes immuables qui ne sont pas mises à jour après leur création. Si des filtres de source sont appliqués à des colonnes mutables qui peuvent être mises à jour après leur création, un comportement négatif peut en résulter.

  Par exemple, un filtre permettant d'exclure ou d'inclure des lignes spécifiques dans une colonne exclut ou inclut toujours les lignes spécifiées, même si les lignes sont modifiées ultérieurement. Supposons que vous excluez ou incluez les lignes 1 à 10 dans la colonne A et qu'elles deviennent plus tard les lignes 11 à 20. Dans ce cas, elles continuent à être exclues ou inclues même lorsque les données ne sont plus les mêmes.

  De même, supposons qu'une ligne en dehors de la portée du filtre soit ultérieurement mise à jour (ou mise à jour et supprimée) et qu'elle soit ensuite exclue ou incluse tel que défini par le filtre. Dans ce cas, il est répliqué sur la cible.

Création de règles de filtre de source en JSON

Vous pouvez créer des filtres sources à l'aide du paramètre JSON `filters` d'une règle de sélection. Le paramètre `filters` spécifie un tableau d'un ou plusieurs objets JSON. Chaque objet a des paramètres qui spécifient le type de filtre source, le nom de colonne et les conditions de filtre. Ces conditions de filtre comprennent un ou plusieurs opérateurs de filtre et des valeurs de filtre.

Le tableau suivant illustre les paramètres utilisés pour spécifier le filtrage de la source dans un objet `filters`.

Paramètre	Valeur
`filter-type`	`source`
`column-name`	Paramètre indiquant le nom de la colonne source à laquelle vous voulez appliquer le filtre. Le nom est sensible à la casse.
`filter-conditions`	Tableau d'un ou plusieurs objets contenant un paramètre `filter-operator` et aucun ou plusieurs paramètres de valeur associés, en fonction de la valeur de `filter-operator` .
`filter-operator`	Paramètre doté de l'une des valeurs suivantes : • `lte` : inférieur ou égal à une valeur • `ste` : inférieur ou égal à une valeur (alias `lte`) • `gte` : supérieur ou égal à une valeur • `eq` : égal à une valeur • `noteq` : pas égal à une valeur • `between` : égal ou compris entre deux valeurs • `notbetween` : pas égal ou compris entre deux valeurs • `null` : valeurs NULL • `notnull` : pas de valeur NULL
`value` ou `start-value` et `end-value` ou pas de valeur	Aucun ou plusieurs paramètres de valeur associés à `filter-operator` : • Si `filter-operator` est défini sur `lte`, `ste`, `gte`, `eq` ou `noteq`, utilisez `value` pour spécifier un paramètre de valeur. • Si `filter-operator` est défini sur `between` ou `notbetween` , utilisez `start-value` et `end-value` pour spécifier deux paramètres de valeur. • Si `filter-operator` est défini sur `null` ou `notnull`, ne spécifiez aucun paramètre de valeur.

Les exemples suivants illustrent des manières courantes d'utiliser des filtres de source.

Example Filtre unique

Le filtre suivant réplique tous les employés où `empid >= 100` dans la base de données cible.

```
{
    "rules": [{
        "rule-type": "selection",
        "rule-id": "1",
        "rule-name": "1",
        "object-locator": {
            "schema-name": "test",
            "table-name": "employee"
        },
        "rule-action": "include",
        "filters": [{
            "filter-type": "source",
            "column-name": "empid",
            "filter-conditions": [{
                "filter-operator": "gte",
                "value": "50"
            },{
                "filter-operator": "noteq",
                "value": "100"
            }]
        }]
    }]
}
```

Example Plusieurs opérateurs de filtre

Le filtre suivant s'applique à plusieurs opérateurs de filtre dans une seule colonne de données. Le filtre réplique tous les employés où `(empid <= 10)` OU `(empid is between 50 and 75)` OU `(empid >= 100)` dans la base de données cible.

```
{
    "rules": [{
        "rule-type": "selection",
        "rule-id": "1",
        "rule-name": "1",
        "object-locator": {
```

```
            "schema-name": "test",
            "table-name": "employee"
        },
        "rule-action": "include",
        "filters": [{
            "filter-type": "source",
            "column-name": "empid",
            "filter-conditions": [{
                "filter-operator": "lte",
                "value": "10"
            }, {
                "filter-operator": "between",
                "start-value": "50",
                "end-value": "75"
            }, {
                "filter-operator": "gte",
                "value": "100"
            }]
        }]
    }]
}
```

Example Plusieurs filtres

Les filtres suivants appliquent plusieurs filtres dans deux colonnes d'une table. Le filtre réplique tous les employés où (`empid <= 100`) ET (`dept = tech`) dans la base de données cible.

```
{
    "rules": [{
        "rule-type": "selection",
        "rule-id": "1",
        "rule-name": "1",
        "object-locator": {
            "schema-name": "test",
            "table-name": "employee"
        },
        "rule-action": "include",
        "filters": [{
            "filter-type": "source",
            "column-name": "empid",
            "filter-conditions": [{
```

```
                "filter-operator": "lte",
                "value": "100"
            }]
        }, {
            "filter-type": "source",
            "column-name": "dept",
            "filter-conditions": [{
                "filter-operator": "eq",
                "value": "tech"
            }]
        }]
    }]
}
```

Example Filtrage des valeurs NULL

Le filtre suivant montre comment filtrer les valeurs vides. Il réplique tous les employés où `dept = NULL` dans la base de données cible.

```
{
    "rules": [{
        "rule-type": "selection",
        "rule-id": "1",
        "rule-name": "1",
        "object-locator": {
            "schema-name": "test",
            "table-name": "employee"
        },
        "rule-action": "include",
        "filters": [{
            "filter-type": "source",
            "column-name": "dept",
            "filter-conditions": [{
                "filter-operator": "null"
            }]
        }]
    }]
}
```

Example Filtrage à l'aide d'opérateurs NOT

Certains opérateurs peuvent être utilisés sous forme négative. Le filtre suivant réplique tous les employés où `(empid is < 50) OR (empid is > 75)` dans la base de données cible.

```
{
    "rules": [{
        "rule-type": "selection",
        "rule-id": "1",
        "rule-name": "1",
        "object-locator": {
            "schema-name": "test",
            "table-name": "employee"
        },
        "rule-action": "include",
        "filters": [{
            "filter-type": "source",
            "column-name": "empid",
            "filter-conditions": [{
                "filter-operator": "notbetween",
                "start-value": "50",
                "end-value": "75"
            }]
        }]
    }]
}
```

Example Utilisation d'opérateurs de filtres mixtes

À partir d'AWS DMS version 3.5.0, vous pouvez mélanger des opérateurs inclusifs et des opérateurs négatifs.

Le filtre suivant réplique tous les employés où `(empid != 50) AND (dept is not NULL)` dans la base de données cible.

```
{
    "rules": [{
        "rule-type": "selection",
        "rule-id": "1",
        "rule-name": "1",
```

```
        "object-locator": {
            "schema-name": "test",
            "table-name": "employee"
        },
        "rule-action": "include",
        "filters": [{
            "filter-type": "source",
            "column-name": "empid",
            "filter-conditions": [{
                "filter-operator": "noteq",
                "value": "50"
            }]
        }, {
            "filter-type": "source",
            "column-name": "dept",
            "filter-conditions": [{
                "filter-operator": "notnull"
            }]
        }]
    }]
}
```

Notez ce qui suit lorsque vous utilisez `null` avec d'autres opérateurs de filtre :

- L'utilisation conjointe de conditions inclusives, négatives et de filtre `null` au sein d'un même filtre ne permet pas de répliquer les enregistrements contenant des valeurs `NULL`.
- L'utilisation conjointe de conditions négatives et de filtre `null` sans conditions de filtre inclusives au sein d'un même filtre ne permet pas de répliquer les données.
- L'utilisation de conditions de filtre négatives sans définir une condition de filtre `null` de manière explicite ne permet pas de répliquer les enregistrements contenant des valeurs `NULL`.

Filtrage par heure et date

Lorsque vous sélectionnez les données à importer, vous pouvez spécifier une date et une heure dans vos critères de filtre. AWS DMS utilise le format de date YYYY-MM-DD et le format d'heure YYYY-MM-DD HH:MM:SS pour le filtre. Les fonctions de comparaison AWS DMS respectent les conventions SQLite. Pour de plus amples informations sur les types de donnée SQLite et les comparaisons de dates, veuillez consulter Types de données dans SQLite Version 3 dans la documentation SQLite.

Le filtre suivant montre comment filtrer sur une date. Il réplique tous les employés où `empstartdate >= January 1, 2002` dans la base de données cible.

Example Filtre de date unique

```
{
    "rules": [{
        "rule-type": "selection",
        "rule-id": "1",
        "rule-name": "1",
        "object-locator": {
            "schema-name": "test",
            "table-name": "employee"
        },
        "rule-action": "include",
        "filters": [{
            "filter-type": "source",
            "column-name": "empstartdate",
            "filter-conditions": [{
                "filter-operator": "gte",
                "value": "2002-01-01"
            }]
        }]
    }]
}
```

Activation et utilisation des évaluations de prémigration pour une tâche

Une évaluation de prémigration évalue les composants spécifiques d'une tâche de migration de base de données et vous aidera à identifier les problèmes susceptibles d'empêcher une tâche de migration de s'exécuter comme prévu. Cette évaluation vous permet d'identifier et de résoudre les problèmes avant d'exécuter une nouvelle tâche ou de la modifier. Cela vous permet d'éviter les retards liés à des échecs de tâches dus à des exigences manquantes ou à des limitations connues.

AWS DMS donne accès à deux options différentes pour les évaluations de prémigration :

- Évaluation du type de données : ancien rapport qui fournit une portée limitée d'évaluations.

- Exécution d'une évaluation préalable à la migration : contient différents types d'évaluations individuelles, y compris les résultats d'évaluation des types de données.

> **Note**
>
> Si vous choisissez une exécution d'évaluation préalable à la migration, il n'est pas nécessaire de choisir une évaluation du type de données séparément.

Ces options sont décrites dans les rubriques suivantes :

- Spécification, démarrage et affichage des exécutions d'évaluation de prémigration: Une évaluation de prémigration (recommandée) spécifie une ou plusieurs évaluations individuelles à exécuter en fonction d'une configuration de tâche de migration nouvelle ou existante. Chaque évaluation individuelle évalue un élément spécifique d'une base de données source et/ou cible prise en charge du point de vue de critères tels que le type de migration, les objets pris en charge, la configuration de l'index et d'autres paramètres de tâches, tels que les mappages de tables identifiant les schémas et les tables à migrer.

  Par exemple, une évaluation individuelle peut évaluer les types de données sources ou les formats de clés primaires qui peuvent ou ne peuvent pas être migrés, éventuellement en fonction de la version AWS DMS du moteur. Vous pouvez démarrer et consulter les résultats de la dernière exécution d'évaluation et consulter les résultats de toutes les séries d'évaluation précédentes pour une tâche, soit à l'aide de la console de AWS DMS gestion, soit à l'aide des SDK AWS CLI et pour accéder à l' AWS DMS API. Vous pouvez également consulter les résultats des évaluations précédentes pour une tâche dans un compartiment Amazon S3 que vous avez sélectionné AWS DMS pour stocker ces résultats.

  > **Note**
  >
  > Le nombre et les types d'évaluations individuelles disponibles peuvent augmenter au fil du temps. Pour plus d'informations sur les mises à jour périodiques, consultez Spécification des évaluations individuelles.

- Démarrage et affichage des évaluations des types de données (Legacy): Une évaluation des types de données (ancienne) renvoie les résultats d'un seul type d'évaluation préalable à la migration dans une structure JSON unique : les types de données susceptibles de ne pas être migrés

correctement dans une instance de base de données source relationnelle prise en charge. Ce rapport renvoie les résultats pour tous les types de données problématiques trouvés dans chaque schéma et chaque table de la base de données source sélectionnés pour la migration.

Création de conditions préalables pour les évaluations préalables à la migration

Cette section décrit les ressources Amazon S3 et IAM dont vous avez besoin pour créer une évaluation préalable à la migration.

Création d'un compartiment S3

AWS DMS stocke les rapports d'évaluation avant la migration dans un compartiment S3. Pour créer le compartiment S3, procédez comme suit :

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Choisissez Créer un compartiment.
3. *Sur la page **Créer un bucket**, entrez un nom unique global incluant votre nom de connexion pour le bucket, tel que dms-bucket- yoursignin.*
4. Choisissez le Région AWS pour la tâche de migration DMS.
5. Laissez les autres paramètres tels quels, puis choisissez Create bucket.

Création de ressources IAM

DMS utilise un rôle et une politique IAM pour accéder au compartiment S3 afin de stocker les résultats des évaluations de prémigration.

Pour créer la politique IAM, procédez comme suit :

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Politiques.
3. Sélectionnez Create policy (Créer une politique).
4. Sur la page Créer une politique, choisissez l'onglet JSON.

5. Collez le code JSON suivant dans l'éditeur en remplaçant l'exemple de code. Remplacez *my-bucket* par le nom du bucket Amazon S3 que vous avez créé dans la section précédente.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:DeleteObject",
            "s3:GetObject",
            "s3:PutObjectTagging"
         ],
         "Resource":[
            "arn:aws:s3:::my-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket",
            "s3:GetBucketLocation"
         ],
         "Resource":[
            "arn:aws:s3:::my-bucket"
         ]
      }
   ]
}
```

6. Choisissez Suivant : Balises, puis cliquez sur Suivant : Réviser.
7. Entrez **DMSPremigrationAssessmentS3Policy** pour Nom*, puis choisissez Créer une politique.

Pour créer le rôle IAM, procédez comme suit :

1. Dans la console IAM, dans le volet de navigation, sélectionnez Rôles.
2. Sélectionnez Créer un rôle.
3. Sur la page Sélectionner une entité de confiance, pour Type d'entité approuvée, choisissez Service AWS . Pour les cas d'utilisation d'autres AWS services, choisissez DMS.

4. Cochez la case DMS, puis choisissez Next.
5. Sur la page Ajouter des autorisations, choisissez DMS PremigrationAssessment S3Policy. Choisissez Suivant.
6. Sur la page Nommer, vérifier et créer, entrez **`DMSPremigrationAssessmentS3Role`** pour Nom du rôle, puis choisissez Créer un rôle.
7. Sur la page Rôles, entrez **`DMSPremigrationAssessmentS3Role`** pour Nom du rôle. Choisissez DMS PremigrationAssessment S3Role.
8. Sur la page DMS PremigrationAssessment S3Role, choisissez l'onglet Relations de confiance. Choisissez Edit trust policy (Modifier la politique d'approbation).
9. Sur la page Modifier la politique d'approbation, collez le code JSON suivant dans l'éditeur en remplaçant le texte existant.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"",
         "Effect":"Allow",
         "Principal":{
            "Service":"dms.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

   Cette politique autorise DMS à placer les résultats de l'exécution de l'évaluation `sts:AssumeRole` préalable à la migration dans le compartiment S3.
10. Choisissez Mettre à jour une politique.

Spécification, démarrage et affichage des exécutions d'évaluation de prémigration

Une évaluation préalable à la migration indique une ou plusieurs évaluations individuelles à exécuter en fonction d'une configuration de tâche de migration nouvelle ou existante. Chaque évaluation individuelle évalue un élément spécifique de la base de données source ou cible en fonction de considérations, telles que le type de migration, les objets pris en charge, la configuration de l'index

et d'autres paramètres de tâche, tels que les mappages de table pour identifier les schémas et les tables à migrer. Par exemple, une évaluation individuelle peut évaluer les types de données sources ou les formats de clés primaires qui peuvent ou ne peuvent pas être migrés.

Spécification des évaluations individuelles

Lorsque vous créez une nouvelle série d'évaluations, vous pouvez choisir d'exécuter certaines ou toutes les évaluations individuelles applicables à la configuration de vos tâches.

AWS DMS prend en charge les exécutions d'évaluation préalable à la migration pour les moteurs de base de données source et cible relationnels suivants :

- Évaluations Oracle
- Évaluations de SQL Server
- Évaluations MySQL(inclut l'édition compatible avec MariaDB et Amazon Aurora MySQL)
- Évaluations PostgreSQL(inclut l'édition compatible avec Amazon Aurora PostgreSQL)

Démarrage et affichage des exécutions d'évaluation de prémigration

Vous pouvez démarrer une évaluation préalable à la migration pour une tâche de migration nouvelle ou existante à l'aide de la console AWS DMS de gestion, de l' AWS CLI API et de l' AWS DMS API.

Pour démarrer une exécution d'évaluation de prémigration pour une tâche nouvelle ou existante

1. À partir de la page Tâches de migration de base de données de la console de gestion AWS DMS , effectuez l'une des actions suivantes :

   - Pour créer une nouvelle tâche et l'évaluer, choisissez Créer une tâche. La page Créer une tâche de migration de base de données s'ouvre :

     1. Entrez les paramètres de tâche requis pour créer votre tâche, y compris le mappage de table.
     2. Dans la section Évaluation de la prémigration, la case à cocher Exécution de l'évaluation de la prémigration est cochée. Cette page contient les options permettant de spécifier une évaluation exécutée pour la nouvelle tâche.

> **Note**
>
> Lors de la création d'une nouvelle tâche, l'activation d'une exécution d'évaluation de prémigration désactive l'option permettant de démarrer automatiquement la tâche lors de sa création. Vous pouvez démarrer la tâche manuellement une fois l'exécution d'évaluation terminée.

- Pour évaluer une tâche existante, choisissez l'identifiant d'une tâche existante sur la page Tâches de migration de base de données. La page de la tâche existante choisie s'ouvre :

  1. Choisissez Actions, puis sélectionnez Créer une évaluation de pré-migration. La page Créer une évaluation de pré-migration s'ouvre avec des options permettant de spécifier une exécution d'évaluation pour la tâche existante.

2. Entrez un nom unique pour votre cycle d'évaluation ou conservez la valeur par défaut.
3. Sélectionnez les évaluations individuelles disponibles que vous souhaitez inclure dans cette exécution d'évaluation. Vous ne pouvez sélectionner que les évaluations individuelles disponibles en fonction de vos paramètres de tâche actuels. Par défaut, toutes les évaluations individuelles disponibles sont activées et sélectionnées.
4. Recherchez et choisissez un compartiment et un dossier Amazon S3 dans votre compte pour stocker le rapport des résultats de votre évaluation. Pour plus d'informations sur la configuration des ressources pour les cycles d'évaluation, consultezCréation de conditions préalables pour les évaluations préalables à la migration.
5. Sélectionnez ou entrez un rôle IAM doté d'un accès complet au compartiment et au dossier Amazon S3 choisis. Pour plus d'informations sur la configuration des ressources pour les cycles d'évaluation, consultezCréation de conditions préalables pour les évaluations préalables à la migration.
6. Choisissez éventuellement un paramètre pour chiffrer le rapport des résultats de l'évaluation dans votre compartiment Amazon S3. Pour plus d'informations sur le chiffrement des compartiments S3, consultez la section Configuration du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.
7. Choisissez Créer une tâche pour une nouvelle tâche ou Créer pour une tâche existante.

   La page Tâches de migration de base de données s'ouvre et répertorie vos tâches nouvelles ou modifiées dont le Statut est Création... ainsi qu'un message de type bannière indiquant que votre exécution d'évaluation de prémigration débutera une fois la tâche créée.

AWS DMS permet d'accéder aux dernières évaluations de prémigration et à toutes les précédentes à l'aide de la console AWS DMS de gestion, de l' AWS CLI API ou de l' AWS DMS API.

Pour afficher les résultats de l'exécution de l'évaluation

1. Dans la console de AWS DMS gestion, choisissez l'identifiant de votre tâche existante sur la page des tâches de migration de base de données. La page de la tâche existante s'ouvre.
2. Choisissez l'onglet Évaluations de pré-migration sur la page de la tâche existante. Cela ouvre une section d'évaluations de prémigration sur cette page affichant les résultats des séries d'évaluation, répertoriés par nom, dans l'ordre chronologique inverse. Le dernier résultat apparaît en haut de la liste. Choisissez le nom de l'exécution d'évaluation dont vous souhaitez consulter les résultats.

Les résultats de cette exécution d'évaluation commencent par le nom de la dernière exécution d'évaluation et un aperçu de son statut, suivis d'une liste des évaluations individuelles spécifiées et de leur statut. Vous pouvez ensuite explorer les détails du statut de chaque évaluation individuelle en choisissant son nom dans la liste. Les résultats sont disponibles jusqu'en bas de la colonne de table.

L'aperçu du statut d'une exécution d'évaluation et chaque évaluation individuelle indiquent une valeur de Statut. Cette valeur indique le statut général de l'exécution d'évaluation et un statut similaire pour chaque évaluation individuelle. Voici une liste des valeurs de Statut pour l'exécution d'évaluation :

- `"cancelling"` : l'exécution d'évaluation a été annulée.
- `"deleting"` : l'exécution d'évaluation a été supprimée.
- `"failed"` : au moins une évaluation individuelle est terminée avec le statut `failed`.
- `"error-provisioning"` : une erreur interne s'est produite lors du provisionnement des ressources (pendant le statut `provisioning`).
- `"error-executing"` : une erreur interne s'est produite lors de l'exécution des évaluations individuelles (pendant le statut `running`).
- `"invalid state"` : l'exécution d'évaluation est dans un état inconnu.
- `"passed"` : toutes les évaluations individuelles sont terminées et aucune n'a le statut `failed`.
- `"provisioning"` : les ressources nécessaires à l'exécution des évaluations individuelles sont en cours de provisionnement.
- `"running"` : les évaluations individuelles sont en cours d'exécution.
- `"starting"` : l'exécution d'évaluation commence, mais les ressources ne sont pas encore provisionnées pour les évaluations individuelles.

Voici une liste des valeurs de Statut pour chaque évaluation individuelle de l'exécution d'évaluation :

- `"cancelled"` : l'évaluation individuelle a été annulée dans le cadre de l'annulation de l'exécution d'évaluation.
- `"error"` : l'évaluation individuelle ne s'est pas terminée avec succès.
- `"failed"` : l'évaluation individuelle s'est terminée avec succès avec un échec de validation : consultez les détails du résultat pour plus d'informations.
- `"invalid state"` : l'évaluation individuelle est dans un état inconnu.
- `"passed"` : l'évaluation individuelle s'est terminée par un résultat de validation réussi.
- `"pending"` : l'évaluation individuelle est en attente d'exécution.
- `"running"` : l'évaluation individuelle est en cours d'exécution.
- `"warning"` : l'évaluation individuelle s'est terminée avec succès avec un avertissement de validation : consultez les détails du résultat pour plus d'informations.

Vous pouvez également afficher les fichiers JSON des résultats de l'exécution d'évaluation sur Amazon S3.

Pour afficher les fichiers JSON de l'exécution d'évaluation sur Amazon S3

1. Dans la console de AWS DMS gestion, choisissez le lien vers le compartiment Amazon S3 affiché dans l'aperçu du statut de l'exécution de l'évaluation. La liste des dossiers de compartiment et des autres objets Amazon S3 stockés dans le compartiment s'affiche. Si vos résultats sont stockés dans un dossier de compartiment, ouvrez-le.
2. Vous trouverez les résultats de votre exécution d'évaluation dans plusieurs fichiers JSON. Un fichier `summary.json` contient les résultats globaux de l'exécution d'évaluation. Les fichiers restants portent chacun le nom d'une évaluation individuelle qui a été spécifiée pour l'exécution d'évaluation, telle que `unsupported-data-types-in-source.json`. Ces fichiers contiennent chacun les résultats de l'évaluation individuelle correspondante à partir de l'exécution d'évaluation choisie.

Pour démarrer et consulter les résultats des évaluations de prémigration pour une tâche de migration existante, vous pouvez exécuter les commandes CLI et les opérations d' AWS DMS API suivantes :

- CLI : `describe-applicable-individual-assessments`, API : `DescribeApplicableIndividualAssessments` – Fournit la liste des évaluations individuelles

que vous pouvez spécifier pour une nouvelle exécution d'évaluation de prémigration, en fonction d'un ou de plusieurs paramètres de configuration de tâche.

- CLI : `start-replication-task-assessment-run`, API : `StartReplicationTaskAssessmentRun` – Démarre une nouvelle exécution d'évaluation de prémigration pour une ou plusieurs évaluations individuelles d'une tâche de migration existante.
- CLI : `describe-replication-task-assessment-runs`, API : `DescribeReplicationTaskAssessmentRuns` – Renvoie une liste paginée des exécutions d'évaluation de prémigration en fonction des paramètres de filtre.
- CLI : `describe-replication-task-individual-assessments`, API : `DescribeReplicationTaskIndividualAssessments` – Renvoie une liste paginée des évaluations individuelles en fonction des paramètres de filtre.
- CLI : `cancel-replication-task-assessment-run`, API : `CancelReplicationTaskAssessmentRun` – Annule, mais ne supprime pas, une seule exécution d'évaluation de prémigration.
- CLI : `delete-replication-task-assessment-run`, API : `DeleteReplicationTaskAssessmentRun` – Supprime l'enregistrement d'une seule exécution d'évaluation de prémigration.

Évaluations individuelles

Cette section décrit les évaluations de prémigration individuelles.

Pour créer une évaluation de prémigration individuelle à l'aide de l' AWS DMS API, utilisez la clé d'API répertoriée pour le `IncludeOnly` paramètre de l' StartReplicationTaskAssessmentRunaction.

Rubriques

- Évaluations pour tous les types de terminaux
- Évaluations Oracle
- Évaluations de SQL Server
- Évaluations MySQL
- Évaluations MariaDB
- Évaluations PostgreSQL

Évaluations pour tous les types de terminaux

Cette section décrit les évaluations de prémigration individuelles pour tous les types de terminaux.

Rubriques

- Types de données non pris en charge
- Les objets de grande taille (LOB) sont utilisés mais les colonnes LOB cibles ne sont pas nullables
- Table source contenant des objets de grande taille (LOB) mais sans clés primaires ni contraintes uniques
- Table source sans clé primaire pour les tâches CDC ou à chargement complet et CDC uniquement
- Table cible sans clés primaires pour les tâches CDC uniquement
- Types de clés primaires source non pris en charge : clés primaires composites

Types de données non pris en charge

Clé API : `unsupported-data-types-in-source`

Vérifie les types de données dans le point de terminaison source qui ne sont pas pris en charge par DMS. Tous les types de données ne peuvent pas être migrés d'un moteur à l'autre.

Les objets de grande taille (LOB) sont utilisés mais les colonnes LOB cibles ne sont pas nullables

Clé API : `full-lob-not-nullable-at-target`

Vérifie la nullabilité d'une colonne LOB dans la cible lorsque la réplication utilise le mode LOB complet ou le mode LOB en ligne. DMS nécessite qu'une colonne LOB soit nulle lors de l'utilisation de ces modes LOB. Cette évaluation nécessite que les bases de données source et cible soient relationnelles.

Table source contenant des objets de grande taille (LOB) mais sans clés primaires ni contraintes uniques

Clé API : `table-with-lob-but-without-primary-key-or-unique-constraint`

Vérifie la présence de tables sources comportant des objets LOB, mais sans clé primaire ni clé unique. Une table doit avoir une clé primaire ou une clé unique pour que DMS puisse migrer les LOB. Cette évaluation nécessite que la base de données source soit relationnelle.

Table source sans clé primaire pour les tâches CDC ou à chargement complet et CDC uniquement

Clé API : `table-with-no-primary-key-or-unique-constraint`

Vérifie la présence d'une clé primaire ou d'une clé unique dans les tables sources pour une migration à chargement complet et à capture des données modifiées (CDC), ou pour une migration uniquement avec CDC. L'absence d'une clé primaire ou d'une clé unique peut entraîner des problèmes de performances lors de la migration vers le CDC. Cette évaluation nécessite que la base de données source soit relationnelle et que le type de migration inclue le CDC.

Table cible sans clés primaires pour les tâches CDC uniquement

Clé API : `target-table-has-unique-key-or-primary-key-for-cdc`

Vérifie la présence d'une clé primaire ou d'une clé unique dans les tables cibles déjà créées pour une migration de CDC uniquement. L'absence de clé primaire ou de clé unique peut entraîner des analyses complètes de la table dans la cible lorsque DMS applique des mises à jour et des suppressions. Cela peut entraîner des problèmes de performances lors de la migration vers le CDC. Cette évaluation nécessite que la base de données cible soit relationnelle et que le type de migration inclue le CDC.

Types de clés primaires source non pris en charge : clés primaires composites

Clé API : `unsupported-source-pk-type-for-elasticsearch-target`

Vérifie la présence de clés primaires composites dans les tables sources lors de la migration vers Amazon OpenSearch Service. La clé primaire de la table source doit se composent d'une seule colonne. Cette évaluation nécessite que la base de données source soit relationnelle et que la base de données cible soit DynamoDB.

> **Note**
>
> DMS prend en charge la migration d'une base de données source vers une cible de OpenSearch service où la clé primaire source est composée de plusieurs colonnes.

Évaluations Oracle

Cette section décrit les évaluations de prémigration individuelles pour les tâches de migration qui utilisent un point de terminaison source Oracle.

> **Note**
>
> Pour utiliser les évaluations de prémigration de cette section, vous devez ajouter les autorisations suivantes à `dms_user` :
>
> ```
> grant select on gv_$parameter to dms_user;
> grant select on v_$instance to dms_user;
> grant select on v_$version to dms_user;
> grant select on gv_$ASM_DISKGROUP to dms_user;
> grant select on gv_$database to dms_user;
> grant select on DBA_DB_LINKS to to dms_user;
> grant select on gv_$log_History to dms_user;
> grant select on gv_$log to dms_user;
> grant select on dba_types to dms_user;
> ```
>
> Pour plus d'informations sur les autorisations lors de l'utilisation d'Oracle en tant que source, consultez Privilèges de compte utilisateur requis sur une source Oracle autogérée pour AWS DMS.

Rubriques

- Vérification de la journalisation supplémentaire au niveau de la base de données
- Validation si le lien de base de données requis a été créé pour la base de données de secours
- Validation Oracle pour le type de données LOB et si le lecteur binaire est configuré
- Validation si la base de données est CDB
- Vérification de l'édition de la base de données Oracle
- Validation de la méthode de CDC Oracle pour DMS
- Validation de la configuration d'Oracle RAC pour DMS

Vérification de la journalisation supplémentaire au niveau de la base de données

Clé API : `oracle-supplemental-db-level`

Cette évaluation de prémigration permet de vérifier si une journalisation supplémentaire minimale est activée au niveau de la base de données. Vous devez activer la journalisation supplémentaire pour utiliser une base de données Oracle en tant que source de migration.

Pour activer la journalisation supplémentaire, utilisez la requête suivante :

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA
```

Pour plus d'informations, consultez Configurez une journalisation supplémentaire.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Validation si le lien de base de données requis a été créé pour la base de données de secours

Clé API : `oracle-validate-standby-dblink`

Cette évaluation préalable à la migration permet de vérifier si Dblink a été créé pour la source de base de données de secours Oracle. AWSDMS_DBLINK est une condition préalable à l'utilisation d'une base de données de secours en tant que source. Lorsque vous utilisez Oracle Standby en tant que source, AWS DMS ne valide pas les transactions ouvertes par défaut.

Pour plus d'informations, consultez Utilisation d'une base de données Oracle autogérée comme source pour AWS DMS.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Validation Oracle pour le type de données LOB et si le lecteur binaire est configuré

Clé API : `oracle-binary-lob-source-validation`

Cette évaluation préalable à la migration permet de vérifier si Oracle LogMiner est utilisé pour un point de terminaison de base de données Oracle version 12c ou ultérieure. AWS DMS ne prend pas en charge Oracle LogMiner pour les migrations de colonnes LOB depuis les bases de données Oracle version 12c. Cette évaluation vérifie également la présence de colonnes LOB et fournit des recommandations appropriées.

Pour configurer votre migration afin de ne pas utiliser Oracle LogMiner, ajoutez la configuration suivante à votre point de terminaison source :

```
useLogMinerReader=N;useBfile=Y;
```

Pour plus d'informations, consultez Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Validation si la base de données est CDB

Clé API : `oracle-validate-cdb`

Cette évaluation de prémigration permet de vérifier si la base de données est une base de données de conteneur. AWS DMS ne prend pas en charge la base de données racine de conteneur à locataires multiples (CDB$ROOT).

> **Note**
>
> Cette évaluation n'est requise que pour Oracle versions 12.1.0.1 ou versions ultérieures. Cette évaluation ne s'applique pas aux versions d'Oracle antérieures à 12.1.0.1.

Pour plus d'informations, consultez Restrictions relatives à l'utilisation d'Oracle comme source pour AWS DMS.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Vérification de l'édition de la base de données Oracle

Clé API : `oracle-check-cdc-support-express-edition`

Cette évaluation de prémigration permet de vérifier si l'édition de la base de données source Oracle est Express Edition. AWS DMS ne prend pas en charge la CDC pour Oracle Express Edition (Oracle Database XE) versions 18.0 et ultérieures.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Validation de la méthode de CDC Oracle pour DMS

Clé API : `oracle-recommendation-cdc-method`

Cette évaluation préalable à la migration valide la génération de journaux redo au cours des sept derniers jours et recommande d'utiliser AWS DMS Binary Reader ou Oracle for LogMiner CDC.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Pour plus d'informations sur le choix de la méthode de CDC à utiliser, consultez Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC.

Validation de la configuration d'Oracle RAC pour DMS

Clé API : `oracle-check-rac`

Cette évaluation de prémigration permet de vérifier si la base de données Oracle est un cluster RAC (Real Application Cluster). Les bases de données RAC doivent être configurées correctement. Si la base de données est basée sur RAC, nous vous recommandons d'utiliser AWS DMS Binary Reader for CDC plutôt qu'Oracle LogMiner.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Pour plus d'informations, consultez Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC.

Évaluations de SQL Server

Cette section décrit les évaluations de prémigration individuelles pour les tâches de migration qui utilisent un point de terminaison source Microsoft SQL Server.

Rubriques

- Vérification si le modèle de récupération de la base de données est simple
- Vérification si les tables dans la portée de la tâche contiennent des colonnes calculées
- Vérification si les tables dans la portée de la tâche ont des index COLUMN STORE

- Vérification si les tables à mémoire optimisée font partie de la portée de la tâche
- Vérification si les tables temporelles font partie de la portée de la tâche
- Vérification si la durabilité différée est activée au niveau de la base de données
- Vérification si la récupération accélérée des données est activée au niveau de la base de données
- Vérification si le mappage de table comporte plus de 10 000 tables avec des clés primaires

Vérification si le modèle de récupération de la base de données est simple

Clé API : `sqlserver-check-for-recovery-model`

Cette évaluation préalable à la migration valide le modèle de restauration du point de terminaison source. AWS DMS nécessite que le modèle de restauration soit configuré pour `Bulk logged` ou `Full` pour une réplication continue.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Pour plus d'informations, consultez Conditions préalables pour l'utilisation de la réplication continue (CDC) à partir d'une source SQL Server.

Vérification si les tables dans la portée de la tâche contiennent des colonnes calculées

Clé API : `sqlserver-check-for-computed-fields`

Cette évaluation préalable à la migration vérifie la présence de colonnes calculées. AWS DMS ne prend pas en charge la réplication des modifications à partir de colonnes calculées par SQL Server.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Pour plus d'informations, consultez Limitations relatives à l'utilisation de SQL Server comme source pour AWS DMS.

Vérification si les tables dans la portée de la tâche ont des index COLUMN STORE

Clé API : `sqlserver-check-for-columnstore-indexes`

Cette évaluation préalable à la migration vérifie la présence de tables avec des index ColumnStore. AWS DMS ne prend pas en charge la réplication des modifications à partir de tables SQL Server avec des index ColumnStore.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Pour plus d'informations, consultez Limitations relatives à l'utilisation de SQL Server comme source pour AWS DMS.

Vérification si les tables à mémoire optimisée font partie de la portée de la tâche

Clé API : `sqlserver-check-for-memory-optimized-tables`

Cette évaluation préalable à la migration vérifie la présence de tables optimisées pour la mémoire. AWS DMS ne prend pas en charge la réplication des modifications à partir de tables optimisées pour la mémoire.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Pour plus d'informations, consultez Limitations relatives à l'utilisation de SQL Server comme source pour AWS DMS.

Vérification si les tables temporelles font partie de la portée de la tâche

Clé API : `sqlserver-check-for-temporal-tables`

Cette évaluation préalable à la migration vérifie la présence de tables temporelles. AWS DMS ne prend pas en charge la réplication des modifications à partir de tables temporelles.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Pour plus d'informations, consultez Limitations relatives à l'utilisation de SQL Server comme source pour AWS DMS.

Vérification si la durabilité différée est activée au niveau de la base de données

Clé API : `sqlserver-check-for-delayed-durability`

Cette évaluation préalable à la migration vérifie la présence d'un retard de durabilité. AWS DMS ne prend pas en charge la réplication des modifications provenant de transactions utilisant une durabilité différée.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Pour plus d'informations, consultez Limitations relatives à l'utilisation de SQL Server comme source pour AWS DMS.

Vérification si la récupération accélérée des données est activée au niveau de la base de données

Clé API : `sqlserver-check-for-accelerated-data-recovery`

Cette évaluation préalable à la migration vérifie la présence d'une restauration accélérée des données. AWS DMS ne prend pas en charge la réplication des modifications à partir de bases de données avec une restauration accélérée des données.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Pour plus d'informations, consultez Limitations relatives à l'utilisation de SQL Server comme source pour AWS DMS.

Vérification si le mappage de table comporte plus de 10 000 tables avec des clés primaires

Clé API : `sqlserver-large-number-of-tables`

Cette évaluation de prémigration vérifie la présence de plus de 10 000 tables avec des clés primaires. Les bases de données configurées avec la réplication Microsoft peuvent rencontrer des échecs de tâche si le nombre de tables comportant des clés primaires est trop élevé.

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Pour plus d'informations sur la configuration de la réplication Microsoft, consultez Capture des modifications de données pour SQL Server autogéré sur site ou sur Amazon EC2.

Évaluations MySQL

Cette section décrit les évaluations de prémigration individuelles pour les tâches de migration qui utilisent un point de terminaison source MySQL.

Rubriques

- Valider si une table utilise un moteur de stockage autre qu'Innodb
- Valider si l'incrémentation automatique est activée sur toutes les tables utilisées pour la migration
- Validez si l'image binlog de la base de données est configurée pour FULL prendre en charge le DMS CDC
- Validez si la base de données source est une MySQL Read-Replica
- Validez si une table comporte des partitions et recommandez target_table_prep_mode les paramètres des tâches à chargement complet
- Valider si DMS prend en charge la version de base de données
- Validez si la base de données cible est configurée pour être définie local_infile sur 1
- Valider si la base de données cible contient des tables avec des clés étrangères
- Valider si les tables sources de l'étendue de la tâche sont soumises à des contraintes de cascade
- Validez si les valeurs de délai d'attente sont appropriées pour une source ou une cible MySQL

Valider si une table utilise un moteur de stockage autre qu'Innodb

Clé API : `mysql-check-table-storage-engine`

Cette évaluation préalable à la migration permet de vérifier si le moteur de stockage utilisé pour une table de la base de données MySQL source est un moteur autre qu'Innodb. DMS crée des tables cibles avec le moteur de stockage InnoDB par défaut. Si vous devez utiliser un moteur de stockage autre qu'InnoDB, vous devez créer manuellement la table dans la base de données cible et configurer votre tâche DMS pour qu'elle `TRUNCATE_BEFORE_LOAD` soit utilisée ou `DO_NOTHING` comme paramètre de tâche à chargement complet. Pour plus d'informations sur les paramètres des tâches de chargement complet, consultezParamètres de tâche de chargement complet.

Pour plus d'informations sur les limites des points de terminaison MySQL, consultezLimitations à l'utilisation d'une base de données MySQL comme source pour AWS DMS.

Valider si l'incrémentation automatique est activée sur toutes les tables utilisées pour la migration

Clé API : `mysql-check-auto-increment`

Cette évaluation préalable à la migration permet de vérifier si l'incrémentation automatique est activée dans les tables sources utilisées dans la tâche. DMS ne migre pas l'attribut AUTO_INCREMENT d'une colonne vers une base de données cible.

Pour plus d'informations sur les limites des points de terminaison MySQL, consultezLimitations à l'utilisation d'une base de données MySQL comme source pour AWS DMS. Pour plus d'informations sur la gestion des colonnes d'identité dans MySQL, consultez Gérer les colonnes IDENTITY dans AWS DMS : Partie 2.

Validez si l'image binlog de la base de données est configurée pour **FULL** prendre en charge le DMS CDC

Clé API : `mysql-check-binlog-image`

Cette évaluation préalable à la migration vérifie si l'image binlog de la base de données source est définie sur. `FULL` Dans MySQL, la `binlog_row_image` variable détermine la manière dont un événement de journal binaire est écrit lors de l'utilisation du `ROW` format. Pour garantir la compatibilité avec le DMS et prendre en charge le CDC, définissez la `binlog_row_image` variable sur`FULL`. Ce paramètre garantit que DMS reçoit suffisamment d'informations pour créer le langage de manipulation de données (DML) complet pour la base de données cible pendant la migration.

Pour définir l'image binlog sur`FULL`, procédez comme suit :

- Pour Amazon RDS, cette valeur est `FULL` par défaut.
- Pour les bases de données hébergées sur site ou sur Amazon EC2, définissez `binlog_row_image` la valeur dans `my.ini` (Microsoft Windows) `my.cnf` ou (UNIX).

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Validez si la base de données source est une MySQL Read-Replica

Clé API : `mysql-check-database-role`

Cette évaluation préalable à la migration vérifie si la base de données source est une réplique lue. Pour activer la prise en charge par le CDC du DMS lorsqu'il est connecté à une réplique en lecture, définissez le `log_slave_updates` paramètre sur`True`. Pour plus d'informations sur l'utilisation d'une base de données MySQL autogérée, consultezUtilisation d'une base de données compatible MySQL autogérée comme source pour AWS DMS.

Pour définir la `log_slave_updates` valeur sur`True`, procédez comme suit :

- Pour Amazon RDS, utilisez le groupe de paramètres de la base de données. Pour plus d'informations sur l'utilisation des groupes de paramètres de base de données RDS, consultez la section Utilisation des groupes de paramètres dans le guide de l'utilisateur Amazon RDS.
- Pour les bases de données hébergées sur site ou sur Amazon EC2, définissez `log_slave_updates` la valeur dans `my.ini` (Microsoft Windows) `my.cnf` ou (UNIX).

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Validez si une table comporte des partitions et recommandez **`target_table_prep_mode`** les paramètres des tâches à chargement complet

Clé API : `mysql-check-table-partition`

Cette évaluation préalable à la migration vérifie la présence de tables avec des partitions dans la base de données source. DMS crée des tables sans partitions sur la cible MySQL. Pour migrer des tables partitionnées vers une table partitionnée sur la cible, vous devez effectuer les opérations suivantes :

- Pré-créez les tables partitionnées dans la base de données MySQL cible.
- Configurez votre tâche DMS pour qu'elle `TRUNCATE_BEFORE_LOAD` soit utilisée ou `DO_NOTHING` comme paramètre de tâche à chargement complet.

Pour plus d'informations sur les limites des points de terminaison MySQL, consultezLimitations à l'utilisation d'une base de données MySQL comme source pour AWS DMS.

Valider si DMS prend en charge la version de base de données

Clé API : `mysql-check-supported-version`

Cette évaluation préalable à la migration vérifie si la version de la base de données source est compatible avec DMS. Le CDC n'est pas pris en charge avec les versions 5.5 ou inférieures d'Amazon RDS MySQL, ni avec les versions de MySQL supérieures à 8.0.x. CDC n'est pris en charge que pour les versions 5.6, 5.7 ou 8.0 de MySQL. Pour plus d'informations sur les versions de MySQL prises en charge, consultezPoints de terminaison sources pour la migration des données.

Validez si la base de données cible est configurée pour être définie **`local_infile`** sur 1

Clé API : `mysql-check-target-localinfile-set`

Cette évaluation préalable à la migration vérifie si le `local_infile` paramètre de la base de données cible est défini sur 1. DMS nécessite que le paramètre « local_infile » soit défini sur 1 lors du chargement complet dans votre base de données cible. Pour plus d'informations, consultez Migration de MySQL vers MySQL à l'aide d'AWS DMS..

Cette évaluation n'est valable que pour une tâche à pleine charge ou à pleine charge et pour une tâche CDC.

Valider si la base de données cible contient des tables avec des clés étrangères

Clé API : `mysql-check-fk-target`

Cette évaluation préalable à la migration vérifie si une tâche à chargement complet ou à chargement complet et une tâche CDC migrant vers une base de données MySQL contiennent des tables avec des clés étrangères. Le paramètre par défaut dans DMS consiste à charger les tables par ordre alphabétique. Les tables comportant des clés étrangères et des contraintes d'intégrité référentielle peuvent entraîner l'échec du chargement, car les tables parent et enfant peuvent ne pas être chargées en même temps.

Pour plus d'informations sur l'intégrité référentielle dans le DMS, consultez la section Utilisation des index, des déclencheurs et des contraintes d'intégrité référentielle dans la rubrique. Améliorer les performances d'une migration AWS DMS

Valider si les tables sources de l'étendue de la tâche sont soumises à des contraintes de cascade

Clé API : `mysql-check-cascade-constraints`

Cette évaluation préalable à la migration vérifie si l'une des tables sources MySQL comporte des contraintes de cascade. Les contraintes en cascade ne sont ni migrées ni répliquées par les tâches DMS, car MySQL n'enregistre pas les modifications liées à ces événements dans le binlog. Bien qu' AWS DMS il ne prenne pas en charge ces contraintes, vous pouvez utiliser des solutions de contournement pour les cibles de base de données relationnelles.

Pour plus d'informations sur la prise en charge des contraintes en cascade et d'autres contraintes, consultez Non-migration des index, des clés étrangères ou des mises à jour ou suppressions en cascade la section Résolution des problèmes de migration dans AWS DMS la rubrique.

Validez si les valeurs de délai d'attente sont appropriées pour une source ou une cible MySQL

Clé API : `mysql-check-network-parameter`

Cette évaluation préalable à la migration vérifie si le point de terminaison MySQL d'une tâche possède les `net_read_timeout` `wait_timeout` paramètres `net_wait_timeout` et définis sur au moins 300 secondes. Cela est nécessaire pour éviter les déconnexions lors de la migration.

Pour plus d'informations, consultez Les connexions à une instance MySQL cible sont déconnectées durant une tâche.

Évaluations MariaDB

Cette section décrit les évaluations de prémigration individuelles pour les tâches de migration qui utilisent un point de terminaison source MariaDB.

Pour créer une évaluation de prémigration individuelle à l'aide de l' AWS DMS API, utilisez la clé d'API répertoriée pour le `Include` paramètre de l' StartReplicationTaskAssessmentRunaction.

Rubriques

- Valider si une table utilise un moteur de stockage autre qu'Innodb
- Valider si l'incrémentation automatique est activée sur toutes les tables utilisées pour la migration
- Validez si le format binlog de la base de données est configuré pour ROW prendre en charge le DMS CDC
- Validez si l'image binlog de la base de données est configurée pour FULL prendre en charge le DMS CDC
- Validez si la base de données source est une MariaDB Read-Replica
- Valider si une table comporte des partitions et recommander TRUNCATE_BEFORE_LOAD ou DO_NOTHING pour les paramètres de tâche à chargement complet
- Valider si DMS prend en charge la version de base de données
- Validez si la base de données cible est configurée pour être définie local_infile sur 1
- Valider si la base de données cible contient des tables avec des clés étrangères
- Valider si les tables sources de l'étendue de la tâche sont soumises à des contraintes de cascade
- Valider si les tables sources de l'étendue de la tâche ont généré des colonnes
- Validez si les valeurs de délai d'expiration sont appropriées pour une source MariaDB
- Validez si les valeurs de délai d'attente sont appropriées pour une cible MariaDB

Valider si une table utilise un moteur de stockage autre qu'Innodb

Clé API : `mariadb-check-table-storage-engine`

Cette évaluation préalable à la migration permet de vérifier si le moteur de stockage utilisé pour une table de la base de données source MariaDB est un moteur autre qu'Innodb. DMS crée des tables cibles avec le moteur de stockage InnoDB par défaut. Si vous devez utiliser un moteur de stockage autre qu'InnoDB, vous devez créer manuellement la table dans la base de données cible et configurer votre tâche DMS pour qu'elle `TRUNCATE_BEFORE_LOAD` soit utilisée ou `DO_NOTHING` comme paramètre de tâche à chargement complet. Pour plus d'informations sur les paramètres des tâches de chargement complet, consultezParamètres de tâche de chargement complet.

Pour plus d'informations sur les limites des terminaux MariaDB, consultez. Limitations à l'utilisation d'une base de données MySQL comme source pour AWS DMS

Valider si l'incrémentation automatique est activée sur toutes les tables utilisées pour la migration

Clé API : `mariadb-check-auto-increment`

Cette évaluation préalable à la migration permet de vérifier si l'incrémentation automatique est activée dans les tables sources utilisées dans la tâche. DMS ne migre pas l'attribut AUTO_INCREMENT d'une colonne vers une base de données cible.

Pour plus d'informations sur les limites des terminaux MariaDB, consultez. Limitations à l'utilisation d'une base de données MySQL comme source pour AWS DMS Pour plus d'informations sur la gestion des colonnes d'identité dans MariaDB, voir Gérer les colonnes IDENTITY AWS DMS dans : Partie 2.

Validez si le format binlog de la base de données est configuré pour **`ROW`** prendre en charge le DMS CDC

Clé API : `mariadb-check-binlog-format`

Cette évaluation préalable à la migration permet de vérifier si le format binlog de la base de données source est configuré pour `ROW` prendre en charge la capture des données de modification (CDC) par DMS.

Pour définir le format binlog sur`ROW`, procédez comme suit :

- Pour Amazon RDS, utilisez le groupe de paramètres de la base de données. Pour plus d'informations sur l'utilisation d'un groupe de paramètres RDS, consultez la section Configuration de la journalisation binaire MySQL dans le guide de l'utilisateur Amazon RDS.

- Pour les bases de données hébergées sur site ou sur Amazon EC2, définissez `binlog_format` la valeur dans `my.ini` (Microsoft Windows) `my.cnf` ou (UNIX).

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Pour plus d'informations sur les serveurs MariaDB auto-hébergés, consultez. Utilisation d'une base de données compatible MySQL autogérée comme source pour AWS DMS

Validez si l'image binlog de la base de données est configurée pour **FULL** prendre en charge le DMS CDC

Clé API : `mariadb-check-binlog-image`

Cette évaluation préalable à la migration vérifie si l'image binlog de la base de données source est définie sur. `FULL` Dans MariaDB, `binlog_row_image` la variable détermine la manière dont un événement de journal binaire est écrit lors de l'utilisation du format. `ROW` Pour garantir la compatibilité avec le DMS et prendre en charge le CDC, définissez la `binlog_row_image` variable sur`FULL`. Ce paramètre garantit que DMS reçoit suffisamment d'informations pour créer le langage de manipulation de données (DML) complet pour la base de données cible pendant la migration.

Pour définir l'image binlog sur`FULL`, procédez comme suit :

- Pour Amazon RDS, cette valeur est `FULL` par défaut.
- Pour les bases de données hébergées sur site ou sur Amazon EC2, définissez `binlog_row_image` la valeur dans `my.ini` (Microsoft Windows) `my.cnf` ou (UNIX).

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Pour plus d'informations sur les serveurs MariaDB auto-hébergés, consultez. Utilisation d'une base de données compatible MySQL autogérée comme source pour AWS DMS

Validez si la base de données source est une MariaDB Read-Replica

Clé API : `mariadb-check-database-role`

Cette évaluation préalable à la migration vérifie si la base de données source est une réplique lue. Pour activer la prise en charge par le CDC du DMS lorsqu'il est connecté à une réplique en lecture, définissez le `log_slave_updates` paramètre sur`True`. Pour plus d'informations sur l'utilisation d'une base de données MySQL autogérée, consultezUtilisation d'une base de données compatible MySQL autogérée comme source pour AWS DMS.

Pour définir la `log_slave_updates` valeur sur`True`, procédez comme suit :

- Pour Amazon RDS, utilisez le groupe de paramètres de la base de données. Pour plus d'informations sur l'utilisation des groupes de paramètres de base de données RDS, consultez la section Utilisation des groupes de paramètres dans le guide de l'utilisateur Amazon RDS.
- Pour les bases de données hébergées sur site ou sur Amazon EC2, définissez `log_slave_updates` la valeur dans `my.ini` (Microsoft Windows) `my.cnf` ou (UNIX).

Cette évaluation n'est valide que pour une migration de chargement complet + CDC, ou pour une migration de CDC uniquement. Cette évaluation n'est pas valide pour une migration de chargement complet uniquement.

Valider si une table comporte des partitions et recommander **`TRUNCATE_BEFORE_LOAD`** ou **`DO_NOTHING`** pour les paramètres de tâche à chargement complet

Clé API : `mariadb-check-table-partition`

Cette évaluation préalable à la migration vérifie la présence de tables avec des partitions dans la base de données source. DMS crée des tables sans partitions sur la cible MariaDB. Pour migrer des tables partitionnées vers une table partitionnée sur la cible, vous devez effectuer les opérations suivantes :

- Pré-créez les tables partitionnées dans la base de données MariaDB cible.
- Configurez votre tâche DMS pour qu'elle `TRUNCATE_BEFORE_LOAD` soit utilisée ou `DO_NOTHING` comme paramètre de tâche à chargement complet.

Pour plus d'informations sur les limites des terminaux MariaDB, consultez. Limitations à l'utilisation d'une base de données MySQL comme source pour AWS DMS

Valider si DMS prend en charge la version de base de données

Clé API : `mariadb-check-supported-version`

Cette évaluation préalable à la migration vérifie si la version de la base de données source est compatible avec DMS. Le CDC n'est pas pris en charge avec les versions 10.4 ou inférieures d'Amazon RDS MariaDB, ni avec les versions de MySQL supérieures à 10.11. Pour plus d'informations sur les versions de MariaDB prises en charge, consultez. Points de terminaison sources pour la migration des données

Validez si la base de données cible est configurée pour être définie **`local_infile`** sur 1

Clé API : `mariadb-check-target-localinfile-set`

Cette évaluation préalable à la migration vérifie si le `local_infile` paramètre de la base de données cible est défini sur 1. DMS nécessite que le paramètre « local_infile » soit défini sur 1 lors du chargement complet dans votre base de données cible. Pour plus d'informations, consultez Migration de MySQL vers MySQL à l'aide d'AWS DMS..

Cette évaluation n'est valable que pour une tâche à chargement complet.

Valider si la base de données cible contient des tables avec des clés étrangères

Clé API : `mariadb-check-fk-target`

Cette évaluation préalable à la migration vérifie si une tâche à chargement complet ou à chargement complet et CDC migrant vers une base de données MariaDB contient des tables avec des clés étrangères. Le paramètre par défaut dans DMS consiste à charger les tables par ordre alphabétique. Les tables comportant des clés étrangères et des contraintes d'intégrité référentielle peuvent entraîner l'échec du chargement, car les tables parent et enfant peuvent ne pas être chargées en même temps.

Pour plus d'informations sur l'intégrité référentielle dans le DMS, consultez la section Utilisation des index, des déclencheurs et des contraintes d'intégrité référentielle dans la rubrique. Améliorer les performances d'une migration AWS DMS

Valider si les tables sources de l'étendue de la tâche sont soumises à des contraintes de cascade

Clé API : `mariadb-check-cascade-constraints`

Cette évaluation préalable à la migration vérifie si l'une des tables sources de MariaDB comporte des contraintes en cascade. Les contraintes en cascade ne sont ni migrées ni répliquées par les tâches DMS, car MariaDB n'enregistre pas les modifications apportées à ces événements dans le binlog. Bien qu' AWS DMS il ne prenne pas en charge ces contraintes, vous pouvez utiliser des solutions de contournement pour les cibles de base de données relationnelles.

Pour plus d'informations sur la prise en charge des contraintes en cascade et d'autres contraintes, consultez Non-migration des index, des clés étrangères ou des mises à jour ou suppressions en cascade la section Résolution des problèmes de migration dans AWS DMS la rubrique.

Valider si les tables sources de l'étendue de la tâche ont généré des colonnes

Clé API : `mariadb-check-generated-columns`

Cette évaluation préalable à la migration vérifie si l'une des tables sources de MariaDB a généré des colonnes. Les tâches DMS ne migrent ni ne répliquent les colonnes générées.

Pour plus d'informations sur la façon de migrer les colonnes générées, consultez???.

Validez si les valeurs de délai d'expiration sont appropriées pour une source MariaDB

Clé API : `mariadb-check-source-network-parameter`

Cette évaluation préalable à la migration vérifie si le point de terminaison source MariaDB d'une tâche possède `net_read_timeout` les `net_wait_timeout` paramètres `wait_timeout` et définis sur au moins 300 secondes. Cela est nécessaire pour éviter les déconnexions lors de la migration.

Pour plus d'informations, consultez Les connexions à une instance MySQL cible sont déconnectées durant une tâche.

Validez si les valeurs de délai d'attente sont appropriées pour une cible MariaDB

Clé API : `mariadb-check-target-network-parameter`

Cette évaluation préalable à la migration vérifie si le point de terminaison cible MariaDB d'une tâche possède `net_read_timeout` les `net_wait_timeout` paramètres `wait_timeout` et définis sur au moins 300 secondes. Cela est nécessaire pour éviter les déconnexions lors de la migration.

Pour plus d'informations, consultez Les connexions à une instance MySQL cible sont déconnectées durant une tâche.

Évaluations PostgreSQL

Cette section décrit les évaluations de prémigration individuelles pour les tâches de migration qui utilisent un point de terminaison source PostgreSQL.

Rubriques

- Valider si la version de la base de données source est prise en charge par DMS pour la migration
- Validez le logical_decoding_work_mem paramètre dans la base de données source
- Vérifiez si la base de données source contient des transactions de longue durée
- Valider le paramètre de base de données source max_slot_wal_keep_size
- Vérifiez si le paramètre de base de données source postgres-check-maxwalsenders est défini pour prendre en charge le CDC.
- Vérifiez si la base de données source est configurée pour PGLOGICAL
- Valider si la clé primaire de la table source est de type LOB
- Valider si la table source possède une clé primaire
- Valider si les transactions préparées sont présentes dans la base de données source
- Validez s'wal_sender_timeoutil est défini sur une valeur minimale requise pour prendre en charge le DMS CDC
- Validez s'wal_levelil est défini sur logique dans la base de données source

Valider si la version de la base de données source est prise en charge par DMS pour la migration

Clé API : `postgres-check-dbversion`

Cette évaluation préalable à la migration vérifie si la version de la base de données source est compatible avec. AWS DMS

Validez le **`logical_decoding_work_mem`** paramètre dans la base de données source

Clé API : `postgres-check-for-logical-decoding-work-mem`

Cette évaluation préalable à la migration recommande de régler le `logical_decoding_work_mem` paramètre sur la base de données source. Dans une base de données hautement transactionnelle où vous pouvez avoir des transactions de longue durée ou de nombreuses sous-transactions, cela peut entraîner une augmentation de la consommation de mémoire pour le décodage logique et la nécessité d'un transfert sur disque. Cela se traduit par une latence élevée de la source DMS lors de la réplication. Dans de tels scénarios, vous devrez peut-être effectuer un réglage`logical_decoding_work_mem`. Ce paramètre est pris en charge dans les versions 13 et supérieures de PostgreSQL.

Vérifiez si la base de données source contient des transactions de longue durée

Clé API : `postgres-check-longrunningtxn`

Cette évaluation préalable à la migration vérifie si la base de données source contient des transactions de longue durée qui ont duré plus de 10 minutes. Le démarrage de la tâche peut échouer, car par défaut, DMS vérifie les transactions ouvertes au démarrage de la tâche.

Valider le paramètre de base de données source **`max_slot_wal_keep_size`**

Clé API : `postgres-check-maxslot-wal-keep-size`

Cette évaluation de prémigration vérifie la valeur configurée pour. `max_slot_wal_keep_size` Lorsqu'elle `max_slot_wal_keep_size` est définie sur une valeur autre que celle par défaut, la tâche DMS peut échouer en raison de la suppression des fichiers WAL requis.

Vérifiez si le paramètre de base de données source **`postgres-check-maxwalsenders`** est défini pour prendre en charge le CDC.

Clé API : `postgres-check-maxwalsenders`

Cette évaluation préalable à la migration vérifie la valeur configurée pour dans la `max_wal_senders` base de données source. Le DMS doit être défini `max_wal_senders` sur une valeur supérieure à 1 pour prendre en charge la capture des données de modification (CDC).

Vérifiez si la base de données source est configurée pour **`PGLOGICAL`**

Clé API : `postgres-check-pglogical`

Cette évaluation préalable à la migration vérifie si la `shared_preload_libraries` valeur est définie `PGLOGICAL` pour `pglogical` soutenir le CDC. Notez que vous pouvez ignorer cette évaluation si vous prévoyez d'utiliser le décodage de test pour la réplication logique.

Valider si la clé primaire de la table source est de type LOB

Clé API : `postgres-check-pk-lob`

Cette évaluation préalable à la migration vérifie si la clé primaire d'une table est du type de données LOB (Large Object). DMS ne prend pas en charge la réplication si la table source possède une colonne LOB comme clé primaire.

Valider si la table source possède une clé primaire

Clé API : `postgres-check-pk`

Cette évaluation préalable à la migration vérifie si des clés primaires existent pour les tables utilisées dans le cadre de la tâche. DMS ne prend pas en charge la réplication pour les tables dépourvues de clés primaires, sauf si l'identité de la réplique est définie `full` sur la table source.

Valider si les transactions préparées sont présentes dans la base de données source

Clé API : `postgres-check-preparedtxn`

Cette évaluation préalable à la migration vérifie si des transactions préparées sont présentes dans la base de données source. La création d'emplacements de réplication peut cesser de répondre si des transactions sont préparées dans la base de données source.

Validez s'**`wal_sender_timeout`**il est défini sur une valeur minimale requise pour prendre en charge le DMS CDC

Clé API : `postgres-check-walsenderstimeout`

Cette évaluation de prémigration vérifie si elle `wal_sender_timeout` est réglée sur un minimum de 10 000 millisecondes (10 secondes). Une tâche DMS avec CDC nécessite un minimum de 10 000 millisecondes (10 secondes) et échoue si la valeur est inférieure à 10 000.

Validez s'**`wal_level`**il est défini sur logique dans la base de données source

Clé API : `postgres-check-wallevel`

Cette évaluation de prémigration vérifie si le paramètre `wal_level` est défini sur logique. Pour que DMS CDC fonctionne, ce paramètre doit être activé dans la base de données source.

Démarrage et affichage des évaluations des types de données (Legacy)

> **Note**
>
> Cette section décrit le contenu existant. Nous vous recommandons d'utiliser des cycles d'évaluation de prémigration, décrits précédemment dansSpécification, démarrage et affichage des exécutions d'évaluation de prémigration.
> Les évaluations des types de données ne sont pas disponibles dans la console. Vous ne pouvez exécuter des évaluations de type de données qu'à l'aide de l'API ou de la CLI, et vous ne pouvez afficher les résultats d'une évaluation de type de données que dans le compartiment S3 de la tâche.

Une évaluation des types de données identifie les types de données d'une base de données source susceptibles de ne pas être migrés correctement car la cible ne les prend pas en charge. Au cours de cette évaluation, AWS DMS lit les schémas de base de données source pour une tâche de migration et crée une liste des types de données de colonne. Il compare ensuite cette liste à une liste prédéfinie de types de données pris en charge par AWS DMS. Si votre tâche de migration comporte des types de données non pris en charge, AWS DMS crée un rapport que vous pouvez consulter pour voir si votre tâche de migration contient des types de données non pris en charge. AWS DMS ne crée pas de rapport si votre tâche de migration ne contient aucun type de données non pris en charge.

AWS DMS prend en charge la création de rapports d'évaluation des types de données pour les bases de données relationnelles suivantes :

- Oracle
- SQL Server
- PostgreSQL
- MySQL
- MariaDB
- Amazon Aurora

Vous pouvez démarrer et consulter un rapport d'évaluation du type de données à l'aide de la CLI et des SDK pour accéder à l' AWS DMS API :

- L'interface de ligne de commande utilise la commande `start-replication-task-assessment` pour démarrer une évaluation du type de données. Elle utilise également la commande `describe-replication-task-assessment-results` pour afficher le dernier rapport d'évaluation du type de données au format JSON.
- L' AWS DMS API utilise l'`StartReplicationTaskAssessment`opération pour démarrer une évaluation du type de données et utilise l'`DescribeReplicationTaskAssessmentResults`opération pour afficher le dernier rapport d'évaluation du type de données au format JSON.

Le rapport d'évaluation du type de données est un fichier JSON unique incluant un résumé qui répertorie les types de données non pris en charge et le nombre de colonnes pour chacun d'eux. Il inclut une liste de structures de données pour chaque type de données non pris en charge, y compris les schémas, les tables et les colonnes contenant le type de données non pris en charge. Vous

pouvez utiliser le rapport pour modifier les types de données source et améliorer le succès de la migration.

Il existe deux niveaux de types de données non pris en charge. Les types de données qui apparaissent dans le rapport comme « non pris en charge » ne peuvent pas faire l'objet d'une migration. Les types de données qui apparaissent dans le rapport comme « partiellement pris en charge » peuvent être convertis en un autre type de données, mais ne pas être migrés comme prévu.

L'exemple suivant montre un exemple de rapport d'évaluation du type de données que vous pouvez consulter.


```
{
    "summary":{
        "task-name":"test15",
        "not-supported":{
            "data-type": [
                "sql-variant"
            ],
            "column-count":3
        },
        "partially-supported":{
            "data-type":[
                "float8",
                "jsonb"
            ],
            "column-count":2
        }
    },
    "types":[
        {
            "data-type":"float8",
            "support-level":"partially-supported",
            "schemas":[
                {
                    "schema-name":"schema1",
                    "tables":[
                        {
                            "table-name":"table1",
                            "columns":[
                                "column1",
                                "column2"
                            ]
                        },
```

```
                    {
                        "table-name":"table2",
                        "columns":[
                            "column3",
                            "column4"
                        ]
                    }
                ]
            },
            {
                "schema-name":"schema2",
                "tables":[
                    {
                        "table-name":"table3",
                        "columns":[
                            "column5",
                            "column6"
                        ]
                    },
                    {
                        "table-name":"table4",
                        "columns":[
                            "column7",
                            "column8"
                        ]
                    }
                ]
            }
        ]
    },
    {
        "datatype":"int8",
        "support-level":"partially-supported",
        "schemas":[
            {
                "schema-name":"schema1",
                "tables":[
                    {
                        "table-name":"table1",
                        "columns":[
                            "column9",
                            "column10"
                        ]
                    },
```

```
                        {
                            "table-name":"table2",
                            "columns":[
                                "column11",
                                "column12"
                            ]
                        }
                    ]
                }
            ]
        }
    ]
}
```

AWS DMS stocke les évaluations de type de données les plus récentes et les précédentes dans un compartiment Amazon S3 créé par AWS DMS votre compte. Le nom du compartiment Amazon S3 est au format suivant, où *customerId* est votre ID de client et *customerDNS* est un identifiant interne.

```
dms-customerId-customerDNS
```

> **Note**
>
> Par défaut, vous pouvez créer jusqu'à 100 compartiments Amazon S3 par compte AWS . Dans la AWS DMS mesure où un bucket est créé dans votre compte, assurez-vous qu'il ne dépasse pas votre limite de bucket. Dans le cas contraire, l'évaluation du type de données échoue.

Tous les rapports d'évaluation du type de données pour une tâche de migration donnée sont stockés dans un dossier de compartiment nommé d'après l'identifiant de la tâche. Le nom de fichier de chaque rapport est la date de l'évaluation du type de données au format yyyy-mm-dd-hh -mm. Vous pouvez afficher et comparer les précédents rapports d'évaluation du type de données à partir de la console de gestion Amazon S3.

AWS DMS crée également un rôle AWS Identity and Access Management (IAM) pour autoriser l'accès au compartiment S3 créé pour ces rapports. Le nom du rôle est `dms-access-for-tasks`. Le rôle utilise la stratégie `AmazonDMSRedshiftS3Role`. Si une ResourceNotFoundFaulterreur se produit lors de l'exécution`StartReplicationTaskAssessment`,

consultez ResourceNotFoundFault la section Dépannage pour plus d'informations sur la création manuelle du `dms-access-for-tasks` rôle.

Exécutions d'évaluation du dépannage

Vous trouverez ci-dessous des rubriques relatives à la résolution des problèmes liés à l'exécution de rapports d'évaluation avec AWS Database Migration Service. Ces rubriques peuvent vous aider à résoudre les problèmes courants.

Rubriques

- ResourceNotFoundFault lors de la course StartReplicationTaskAssessment

ResourceNotFoundFault lors de la course StartReplicationTaskAssessment

Vous pouvez rencontrer l'exception suivante lors de l'exécution de l'StartReplicationTaskAssessmentaction.

```
An error occurred (ResourceNotFoundFault) when calling the
 StartReplicationTaskAssessment operation: Task assessment has not been run or dms-
access-for-tasks IAM Role not configured correctly
```

Si vous rencontrez cette exception, créez le dms-access-for-tasksrôle en procédant comme suit :

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Rôles.
3. Sélectionnez Create role (Créer un rôle).
4. Sur la page Sélectionner une entité de confiance, pour Type d'entité fiable, choisissez Politique de confiance personnalisée.
5. Collez le code JSON suivant dans l'éditeur en remplaçant le texte existant.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Principal": {
                "Service": "dms.amazonaws.com"
```

```
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

La politique précédente accorde l'`sts:AssumeRole`autorisation de AWS DMS. Lorsque vous ajoutez la politique AmazondMSRedShifts3Role, DMS peut créer le compartiment S3 dans votre compte et y placer les résultats de l'évaluation du type de données.

6. Choisissez Suivant.
7. Sur la page Ajouter des autorisations, recherchez et ajoutez la politique AmazondMsRedShifts3Role. Choisissez Suivant.
8. Sur la page Nom, révision et création, nommez le rôle dms-access-for-tasks. Sélectionnez Créer un rôle.

Spécification de données supplémentaires pour les paramètres de tâche

Lorsque vous créez ou modifiez une tâche de réplication pour certains points de terminaison AWS DMS, la tâche peut nécessiter des informations supplémentaires pour effectuer la migration. Vous pouvez spécifier ces informations supplémentaires à l'aide d'une option de la console DMS. Ou vous pouvez le spécifier en utilisant le paramètre `TaskData` de l'opération d'API DMS `CreateReplicationTask` ou `ModifyReplicationTask`.

Si votre point de terminaison cible est Amazon Neptune, vous devez spécifier des données de mappage, en complément du mappage de tables. Ces données de mappage supplémentaires spécifient comment convertir les données relationnelles sources en données de graphe cibles qu'une base de données Neptune peut consommer. Dans ce cas, vous pouvez utiliser l'un des deux formats possibles. Pour de plus amples informations, veuillez consulter Spécification de règles de mappage de graphe à l'aide de Gremlin et de R2RML pour Amazon Neptune en tant que cible.

Surveillance des tâches AWS DMS

La surveillance est un enjeu important pour assurer la fiabilité, la disponibilité et les performances d'AWS DMS et de vos solutions AWS. Vous devez collecter les données de surveillance de l'ensemble de votre solution AWS afin de pouvoir déboguer plus facilement une éventuelle défaillance à plusieurs points. AWS fournit plusieurs outils pour surveiller vos ressources AWS DMS et autres, et répondre aux incidents potentiels.

Événements et notifications AWS DMS

AWS DMS utilise Amazon Simple Notification Service (Amazon SNS) pour fournir des notifications lorsqu'un événement AWS DMS se produit, tel que la création ou la suppression d'une instance de réplication. AWS DMS regroupe les événements en catégories auxquelles vous pouvez vous abonner pour être averti lorsqu'un événement de cette catégorie se produit. Par exemple, si vous vous abonnez à la catégorie de création d'une instance de réplication donnée, vous recevez une notification chaque fois que survient un événement lié à la création qui affecte votre instance de réplication. Vous pouvez utiliser ces notifications sous n'importe quelle forme prise en charge par Amazon SNS pour une région AWS, telle qu'un e-mail, un message texte ou un appel à un point de terminaison HTTP. Pour de plus amples informations, veuillez consulter Utilisation des événements et des notifications Amazon SNS dans AWS Database Migration Service.

État de la tâche

Vous pouvez surveiller la progression d'une tâche en vérifiant le statut de la tâche et en surveillant la table de contrôle de la tâche. L'état de la tâche indique l'état d'une tâche AWS DMS et les ressources qui lui sont associées. Il inclut des indications qui permettent de déterminer si la tâche est en cours de création, de lancement, d'exécution ou d'arrêt. Il indique également l'état actuel des tables migrées par la tâche. Par exemple, il permet de déterminer si une charge complète d'une table a commencé ou si elle est en cours et donne des informations telles que le nombre d'insertions, de suppressions et de mises à jour effectuées pour cette table. Pour plus d'informations sur la surveillance de l'état des tâches et de leurs ressources, consultez État de la tâche et État d'une table pendant des tâches. Pour plus d'informations sur les tables de contrôle, consultez Paramètres de tâche de la table de contrôle.

Alarmes et journaux Amazon CloudWatch

Utilisez les alarmes Amazon CloudWatch pour surveiller une ou plusieurs métriques de tâche sur une période que vous spécifiez. Si une métrique dépasse un seuil donné, une notification est envoyée à une rubrique Amazon SNS. Les alarmes CloudWatch n'appellent pas une action

uniquement parce qu'elles se trouvent dans un état particulier. Au contraire, l'état doit avoir changé et avoir été maintenu pendant un nombre déterminé de périodes. AWS DMS utilise également CloudWatch pour journaliser les informations sur les tâches au cours du processus de migration. Vous pouvez utiliser l'AWS CLI ou l'API AWS DMS pour consulter des informations sur les journaux de tâches. Pour plus d'informations sur l'utilisation de CloudWatch avec AWS DMS, consultez Surveillance des tâches de réplication à l'aide d'Amazon CloudWatch. Pour plus d'informations sur la surveillance des métriques AWS DMS, consultez Métriques AWS Database Migration Service. Pour plus d'informations sur l'utilisation des journaux de tâches AWS DMS, consultez Affichage et gestion des journaux de tâches AWS DMS.

Journaux de voyage dans le temps

Pour journaliser et déboguer les tâches de réplication, vous pouvez utiliser le voyage dans le temps AWS DMS. Dans cette approche, vous utilisez Amazon S3 pour stocker les journaux et les chiffrer à l'aide de vos clés de chiffrement. Vous pouvez récupérer vos journaux S3 à l'aide de filtres de date et d'heure, puis afficher, télécharger et masquer les journaux selon vos besoins. Ce faisant, vous pouvez « voyager dans le temps » pour examiner les activités de base de données.

Vous pouvez utiliser le voyage dans le temps avec des points de terminaison sources PostgreSQL pris en charge par DMS et des points de terminaison cibles PostgreSQL et MySQL pris en charge par DMS. Vous pouvez activer le voyage dans le temps uniquement pour les tâches de chargement complet et CDC et pour les tâches de CDC uniquement. Pour activer le voyage dans le temps ou pour modifier des paramètres de voyage dans le temps existants, assurez-vous que votre tâche est arrêtée.

Pour plus d'informations sur les journaux de voyage dans le temps, consultez Paramètres de tâche de voyage dans le temps. Pour examiner les bonnes pratiques relatives à l'utilisation des journaux de voyage dans le temps, consultez Résolution des problèmes liés aux tâches de réplication à l'aide du voyage dans le temps.

AWS CloudTrailJournaux

AWS DMS est intégré avec AWS CloudTrail, un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle IAM ou un service AWS dans AWS DMS. CloudTrail capture tous les appels d'API pour AWS DMS en tant qu'événements, y compris les appels émis par la console AWS DMS et les appels de code transmis aux opérations d'API AWS DMS. Si vous créez un journal d'activité, vous pouvez activer la livraison continue d'événements CloudTrail à un compartiment Amazon S3, y compris des événements pour AWS DMS. Si vous ne configurez pas de journal de suivi, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Event history (Historique des événements). En utilisant les informations

collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée à AWS DMS, l'adresse IP, l'auteur et la date de la demande, ainsi que d'autres détails. Pour de plus amples informations, veuillez consulter Journalisation des appels d'API AWS DMS avec AWS CloudTrail.

Journaux de base de données

Vous pouvez afficher, télécharger et surveiller les journaux de base de données pour vos points de terminaison de tâche à l'aide de la console AWS Management Console, de l'AWS CLI ou de l'API de votre service de base de données AWS. Pour plus d'informations, consultez la documentation de votre service de base de données dans la documentation AWS.

Pour plus d'informations, consultez les rubriques suivantes.

Rubriques

- État de la tâche
- État d'une table pendant des tâches
- Surveillance des tâches de réplication à l'aide d'Amazon CloudWatch
- Métriques AWS Database Migration Service
- Affichage et gestion des journaux de tâches AWS DMS
- Journalisation des appels d'API AWS DMS avec AWS CloudTrail
- Journalisation du contexte AWS DMS

État de la tâche

Le statut de la tâche indique la condition de la tâche. Le tableau suivant montre les états possibles qu'une tâche peut avoir :

État de la tâche	Description
Création	AWS DMS crée la tâche.
En cours d'exécution	La tâche exécute les tâches de migration spécifiées.
Arrêté(e)	La tâche est arrêtée.
Stopping	La tâche est en cours d'arrêt. Ceci est généralement une indication d'une intervention d'un utilisateur dans la tâche.

État de la tâche	Description
Suppression en cours	La tâche est cours de suppression, généralement suite à une demande d'intervention de l'utilisateur.
Échec	La tâche a échoué. Pour plus d'informations, consultez les fichiers journaux de la tâche.
Error (Erreur)	La tâche s'est arrêtée en raison d'une erreur. Une brève description de l'erreur concernant la tâche est fournie dans la dernière section du message d'échec, dans l'onglet Vue d'ensemble.
Exécution avec des erreurs	La tâche s'exécute avec un statut d'erreur. Cela indique généralement qu'une ou plusieurs tables de la tâche n'ont pas pu être migrées. La tâche continue de charger d'autres tables conformément aux règles de sélection.
Démarrage en cours	La tâche se connecte à l'instance de réplication et aux points de terminaison source et cible. Les filtres et les transformations sont appliqués.
Prêt	La tâche est prête à s'exécuter. Cet état suit généralem ent l'état « Création ».
Modification	La tâche est en cours de modification, généralement en raison d'une action de l'utilisateur qui a modifié les paramètres de la tâche.
Déplacement	La tâche est en cours de déplacement vers une autre instance de réplication. La réplication conserve ce statut jusqu'à ce que le déplacement soit terminé. La suppressi on de la tâche est la seule opération autorisée sur la tâche de réplication pendant son déplacement.

État de la tâche	Description
Échec du déplacement	Le déplacement de la tâche a échoué pour une raison quelconque, telle qu'un manque d'espace de stockage sur l'instance de réplication cible. Lorsqu'une tâche de réplication présente ce statut, elle peut être démarrée, modifiée, déplacée ou supprimée.
Test	La migration de base de données spécifiée pour cette tâche est testée en réponse à l'exécution de l'opérati on StartReplicationTaskAssessmentRun ou StartRepl icationTaskAssessment.

La barre d'état des tâches donne une estimation de la progression de la tâche. La qualité de cette estimation dépend de la qualité des statistiques de table de la base de données source ; meilleures sont les statistiques de table, plus précise sera l'estimation. Pour les tâches avec une seule table ne disposant pas de statistiques de lignes estimées, nous ne pouvons par fournir une estimation complète en pourcentage. Dans ce cas, l'état de la tâche et l'indication des lignes chargées peuvent servir à confirmer que la tâche est bien en cours d'exécution et de progression.

Notez que la colonne « Dernière mise à jour » de la console DMS indique uniquement l'instant où AWS DMS a mis à jour pour la dernière fois l'enregistrement des statistiques de table pour une table. Elle n'indique pas l'heure de la dernière mise à jour de la table.

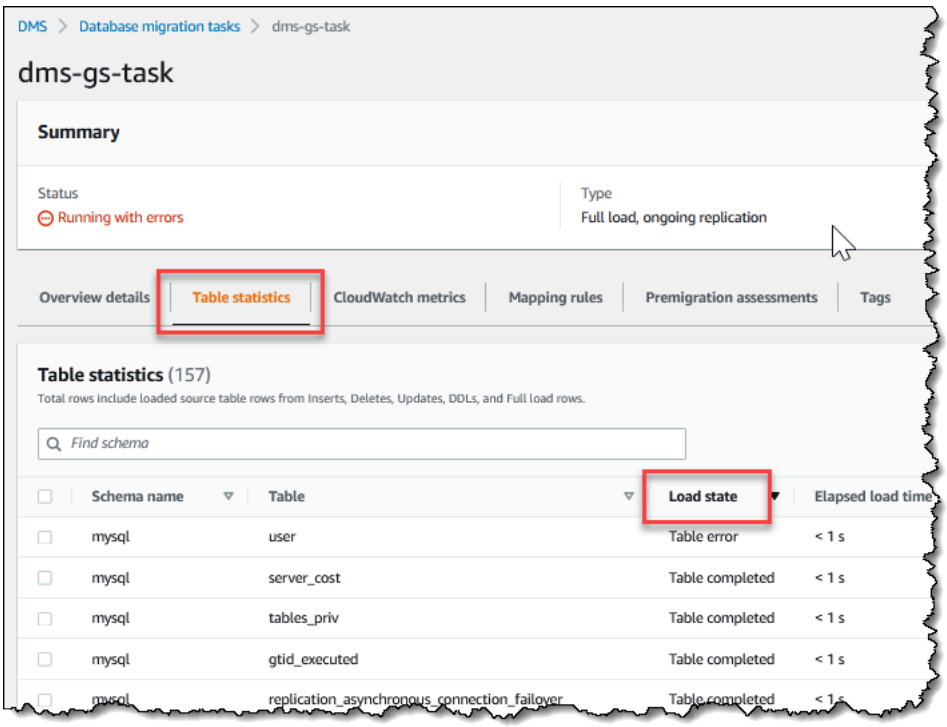
Outre l'utilisation de la console DMS, vous pouvez générer une description des tâches de réplication actuelles, y compris leur statut, à l'aide de la commande `aws dms describe-replication-tasks` dans AWS CLI, comme illustré dans l'exemple suivant.

```
{
    "ReplicationTasks": [
        {
            "ReplicationTaskIdentifier": "moveit2",
            "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
            "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
            "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
            "MigrationType": "full-load",
```

```
            "TableMappings": ...output omitted... ,
            "ReplicationTaskSettings": ...output omitted... ,
            "Status": "stopped",
            "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
            "ReplicationTaskCreationDate": 1590524772.505,
            "ReplicationTaskStartDate": 1590619805.212,
            "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII",
            "ReplicationTaskStats": {
                "FullLoadProgressPercent": 100,
                "ElapsedTimeMillis": 0,
                "TablesLoaded": 0,
                "TablesLoading": 0,
                "TablesQueued": 0,
                "TablesErrored": 0,
                "FreshStartDate": 1590619811.528,
                "StartDate": 1590619811.528,
                "StopDate": 1590619842.068
            }
        }
    ]
}
```

État d'une table pendant des tâches

La console AWS DMS met à jour les informations concernant l'état de vos tables au cours de la migration. Le tableau suivant illustre les valeurs d'état possibles :



État	Description
La table n'existe pas	AWS DMS ne peut pas trouver la table sur le point de terminaison source.
Avant chargement	Le processus de chargement complet a été activé, mais il n'a pas encore commencé.
Chargement complet	Le processus de chargement complet est en cours.
Table terminée	Le chargement complet est terminé.
Table annulée	Le chargement de la table a été annulé.
Erreur de table	Une erreur s'est produite lors du chargement de la table.

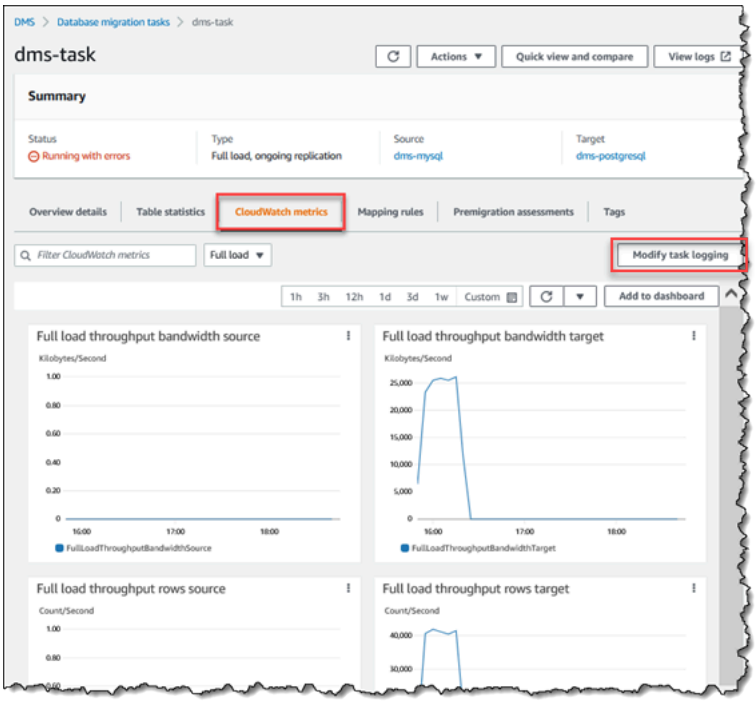
Surveillance des tâches de réplication à l'aide d'Amazon CloudWatch

Vous pouvez utiliser les alarmes ou les événements Amazon CloudWatch pour suivre plus précisément la migration. Pour plus d'informations sur Amazon CloudWatch, consultez Que sont Amazon CloudWatch, Amazon CloudWatch Events et Amazon CloudWatch Logs ? dans le Guide de l'utilisateur Amazon CloudWatch. Notez que l'utilisation d'Amazon CloudWatch implique des frais supplémentaires.

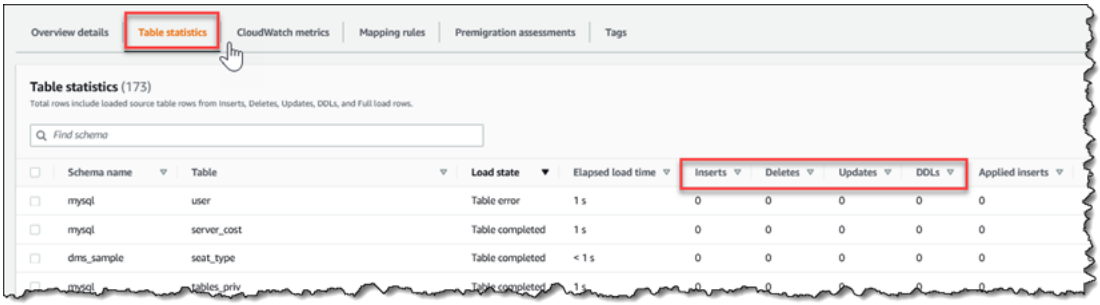
Si votre tâche de réplication ne crée pas de journaux CloudWatch, consultez AWS DMSne crée pas de CloudWatch journaux dans le guide de résolution des problèmes.

La console AWS DMS affiche les statistiques CloudWatch de base pour chaque tâche, notamment le statut, le pourcentage de réalisation, le temps écoulé et les statistiques de table, comme illustré ci-après. Sélectionnez la tâche de réplication, puis cliquez sur l'onglet Métriques CloudWatch.

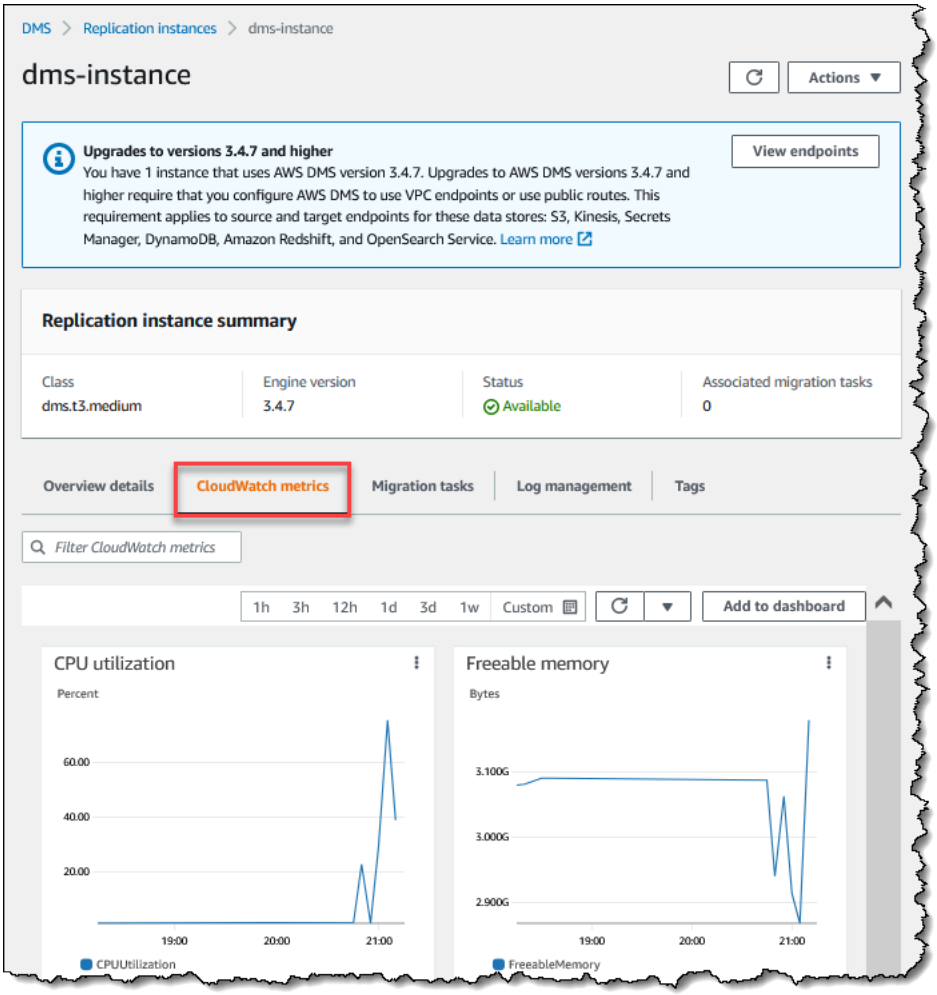
Pour afficher et modifier les paramètres des journaux de tâches CloudWatch, choisissez Modifier la journalisation des tâches. Pour de plus amples informations, veuillez consulter Paramètres de la tâche de journalisation.



La console AWS DMS affiche les statistiques de performances pour chaque table, notamment le nombre d'insertions, de suppressions et de mises à jour, lorsque vous sélectionnez l'onglet Statistiques de table.



En outre, si vous sélectionnez une instance de réplication dans la page Instance de réplication, vous pouvez afficher les métriques de performance pour l'instance en choisissant l'onglet Métriques CloudWatch.



Métriques AWS Database Migration Service

AWS DMS fournit des statistiques pour les éléments suivants :

- Métriques de l'hôte : statistiques de performances et d'utilisation pour l'hôte de réplication, fournies par Amazon CloudWatch. Pour consulter une liste complète des métriques disponibles, consultez la page Métriques des instances de réplication.
- Métriques de la tâche de réplication : statistiques pour les tâches de réplication, notamment les modifications entrantes et validées, ainsi que la latence entre l'hôte de réplication et les bases de données source et cible. Pour consulter une liste complète des métriques disponibles, consultez la page Métriques de tâches de réplication.
- Métriques de table : statistiques pour les tables en cours de migration, notamment le nombre d'instructions d'insertion, de mise à jour, de suppression et DDL terminées.

Les métriques de tâche sont divisées en statistiques entre l'hôte de réplication et le point de terminaison source et en statistiques entre l'hôte de réplication et le point de terminaison cible. Vous pouvez déterminer les statistiques totales d'une tâche en additionnant deux statistiques associées. Par exemple, vous pouvez déterminer la latence totale, ou attente de réplique, d'une tâche, en combinant les valeurs de CDCLatencySource et de CDCLatencyTarget.

Les valeurs de métriques de tâche peuvent être influencées par l'activité en cours dans votre base de données source. Par exemple, si une transaction a commencé mais n'a pas été validée, la métrique CDCLatencySource continue à augmenter jusqu'à ce que cette transaction soit validée.

Pour l'instance de réplication, la métrique FreeableMemory nécessite une clarification. La mémoire libérable n'est pas une indication relative à la mémoire réelle disponible. Il s'agit de la mémoire actuellement en cours d'utilisation et pouvant être libérée et affectée à d'autres utilisations ; il s'agit d'une combinaison des mémoires tampon et de cache en cours d'utilisation sur l'instance de réplication.

Bien que la métrique FreeableMemory ne reflète pas la mémoire réelle disponible, la combinaison des métriques FreeableMemory et SwapUsage peut indiquer si l'instance de réplication est surchargée.

Surveillez les conditions suivantes pour ces deux métriques.

- La métrique FreeableMemory approche la valeur zéro.
- La métrique SwapUsage augmente ou fluctue.

Si vous rencontrez une de ces deux conditions, cela indique que vous devez envisager un transfert vers une instance de réplication plus importante. Vous devez également envisager de

réduire le nombre et le type de tâches exécutées sur l'instance de réplication. Les tâches de chargement complet nécessitent davantage de mémoire que les tâches qui répliquent seulement des modifications.

Pour estimer approximativement les besoins réels en mémoire pour une tâche de migration AWS DMS, vous pouvez utiliser les paramètres suivants.

Colonnes LOB

Nombre moyen de colonnes LOB dans chaque table de votre étendue de migration.

Nombre maximum de tables à charger en parallèle

Nombre maximal de tables chargées par AWS DMS en parallèle dans une tâche.

La valeur par défaut est 8.

Taille de bloc du LOB

Taille des blocs du LOB, en kilo-octets, utilisée par AWS DMS pour répliquer les données vers la base de données cible.

Taux de validation pendant le chargement complet

Nombre maximal d'enregistrements qu'AWS DMS peut transférer en parallèle.

La valeur par défaut est 10,000.

Taille de LOB

Taille maximale d'un LOB individuel, en kilo-octets.

Taille de tableau en masse

Nombre maximal de lignes extraites ou traitées par votre pilote de point de terminaison. Cette valeur dépend des paramètres du pilote.

La valeur par défaut est 1,000.

Après avoir déterminé ces valeurs, vous pouvez utiliser l'une des méthodes suivantes pour estimer la quantité de mémoire requise pour votre tâche de migration. Ces méthodes dépendent de l'option que vous choisissez pour Paramètres de la colonne LOB dans votre tâche de migration.

- Pour Mode LOB complet, utilisez la formule suivante.

```
Required memory = (LOB columns) * (Maximum number of tables to load in
parallel) * (LOB chunk size) * (Commit rate during full load)
```

Prenons un exemple où vos tables sources incluent en moyenne 2 colonnes LOB et où la taille des blocs du LOB est de 64 Ko. Si vous utilisez les valeurs par défaut pour `Maximum number of tables to load in parallel` et `Commit rate during full load`, la quantité de mémoire requise pour votre tâche est la suivante.

```
Required memory = 2 * 8 * 64 * 10,000 = 10,240,000 KB
```

> **Note**
>
> Pour réduire la valeur de Taux de validation (opérations Commit) lors du chargement complet, ouvrez la console AWS DMS, choisissez Tâches de migration de base de données, puis créez ou modifiez une tâche. Développez Paramètres avancés et entrez votre valeur pour Taux de validation (opérations Commit) lors du chargement complet.

- Pour Mode LOB limité, utilisez la formule suivante.

  ```
  Required memory = (LOB columns) * (Maximum number of tables to load in
  parallel) * (LOB size) * (Bulk array size)
  ```

  Prenons un exemple où vos tables sources incluent en moyenne 2 colonnes LOB et où la taille maximale d'un LOB individuel est de 4 096 Ko. Si vous utilisez les valeurs par défaut pour `Maximum number of tables to load in parallel` et `Bulk array size`, la quantité de mémoire requise pour votre tâche est la suivante.

  ```
  Required memory = 2 * 8 * 4,096 * 1,000 = 65,536,000 KB
  ```

Pour qu'AWS DMS effectue les conversions de manière optimale, le CPU doit être disponible au moment des conversions. Une surcharge du CPU et un manque de ressources CPU peuvent ralentir les migrations. AWS DMS peut nécessiter une utilisation intensive de CPU, en particulier lors de migrations et de réplications hétérogènes, telles qu'une migration d'Oracle vers PostgreSQL. L'utilisation d'une classe d'instances de réplication C4 peut être un choix judicieux pour de telles situations. Pour de plus amples informations, veuillez consulter Choisir l'instance de réplication AWS DMS adaptée à votre migration.

Métriques des instances de réplication

La surveillance des instances de réplication comprend les métriques Amazon CloudWatch pour les statistiques suivantes.

Métrique	Description
AvailableMemory	Estimation de la quantité de mémoire disponible pour démarrer de nouvelles applications, sans échange. Pour plus d'informations, consultez la valeur `MemAvailable` dans la section `/proc/meminfo` des pages de manuel Linux. Unités : octets
CPUAllocated	Pourcentage de CPU alloué au maximum pour la tâche (0 signifie qu'il n'y a pas de limite). AWS DMS augmente cette métrique par rapport aux dimensions combinées de `ReplicationInstanceIdentifer` et `ReplicationTaskIdentifier` dans la console CloudWatch. Utilisez la catégorie `ReplicationInstanceIdentifier, ReplicationTaskIdentifier` pour visualiser cette métrique. Unités : pourcentage
CPUUtilization	Pourcentage de vCPU (CPU virtuel) alloué actuellement utilisé dans l'instance. Unités : pourcentage
DiskQueueDepth	Nombre de demandes de lecture et d'écriture (E/S) en attente d'accéder au disque. Unités : nombre
FreeStorageSpace	Quantité d'espace de stockage disponible. Unités : octets

Métrique	Description
FreeMemory	Quantité de mémoire physique disponible pour être utilisée par les applications, le cache de pages et pour les propres structures de données du noyau. Pour plus d'informations, consultez la valeur `MemFree` dans la section `/proc/memInfo` des pages de manuel Linux. Unités : octets
FreeableMemory	Quantité de mémoire vive disponible. Unités : octets
MemoryAllocated	Allocation maximale de mémoire pour la tâche (0 signifie qu'il n'y a pas de limite). AWS DMS augmente cette métrique par rapport aux dimension s combinées de `ReplicationInstanceIdentifer` et `ReplicationTaskIdentifier` dans la console CloudWatch. Utilisez la catégorie `ReplicationInstanceIdentifier, ReplicationTaskIdentifier` pour visualiser cette métrique. Unités : Mio
WriteIOPS	Nombre moyen d'opérations d'I/O d'écriture de disque par seconde. Unités : nombre/seconde
ReadIOPS	Nombre moyen d'opérations d'I/O de lecture de disque par seconde. Unités : nombre/seconde
WriteThroughput	Nombre moyen d'octets écrits sur le disque par seconde. Unités : octets/seconde
ReadThroughput	Nombre moyen d'octets lus sur le disque par seconde. Unités : octets/seconde

Métrique	Description
WriteLatency	Temps moyen nécessaire pour les opérations d'E/S (sortie) par disque. Unités : millisecondes
ReadLatency	Temps moyen nécessaire pour les opérations d'E/S (entrée) par disque. Unités : millisecondes
SwapUsage	Quantité d'espace d'échange utilisé sur l'instance de réplication. Unités : octets
NetworkTransmitThroughput	Trafic réseau sortant (transmission) sur l'instance de réplication, comprenant le trafic de base de données client et le trafic AWS DMS, utilisé pour la surveillance et la réplication. Unités : octets/seconde
NetworkReceiveThroughput	Trafic réseau entrant (réception) sur l'instance de réplication, notamment le trafic de base de données client et le trafic AWS DMS, utilisé pour la surveillance et la réplication. Unités : octets/seconde

Métriques de tâches de réplication

La surveillance de la tâche de réplication inclut des métriques pour les statistiques suivantes.

Métrique	Description
FullLoadThroughputBandwidthTarget	Données sortantes transmises à partir d'un chargement complet pour la cible en Ko par seconde.
FullLoadThroughputRowsTarget	Modifications sortantes à partir d'un chargement total depuis la source, exprimées en lignes par seconde.

Métrique	Description
CDCIncomingChanges	Nombre total d'événements de modification qui attendent, à un moment donné, d'être appliqués à la cible. Notez que cela est différent de la mesure du taux de modifications de transaction du point de terminais on source. Une valeur élevée pour cette métrique indique généralem ent qu'AWS DMS n'est pas en mesure d'appliquer les modifications capturées dans un délai raisonnable, ce qui entraîne une latence cible importante.
CDCChange sMemorySource	Quantité de lignes s'accumulant dans une mémoire et attendant leur validation à partir de la source. Vous pouvez visualiser cette métrique avec CDCChangesDiskSource.
CDCChange sMemoryTarget	Quantité de lignes s'accumulant dans une mémoire et attendant leur validation dans la cible. Vous pouvez visualiser cette métrique avec CDCChangesDiskTarget.
CDCChangesDiskSour ce	Quantité de lignes s'accumulant sur un disque et attendant leur validation à partir de la source. Vous pouvez visualiser cette métrique avec CDCChangesMemorySource.
CDCChangesDiskTarg et	Quantité de lignes s'accumulant sur un disque et attendant leur validation dans la cible. Vous pouvez visualiser cette métrique avec CDCChangesMemoryTarget.
CDCThroughputBandw idthTarget	Données sortantes transmises pour la cible en Ko par seconde. CDCThroughputBandwidth enregistre les données sortantes transmise s sur les points d'échantillonnage. Si aucun trafic réseau de tâche n'est trouvé, la valeur est zéro. Étant donné que CDC ne délivre pas de transactions de longue durée, le trafic réseau peut ne pas être enregistré.
CDCThroughputRowsS ource	Modifications de tâche entrante à partir de la source en lignes par seconde.
CDCThroughputRowsT arget	Modifications de tâche sortante pour la cible en lignes par seconde.

Métrique	Description
CDCLatencySource	Intervalle, en secondes, entre le dernier événement capturé à partir du point de terminaison source et l'horodatage système actuel de l'instanc e AWS DMS. CDCLatencySource représente la latence entre la source et l'instance de réplication. Une valeur élevée de CDCLatencySource signifie que le processus de capture des modifications depuis la source est retardé. Pour identifier la latence dans une réplication continue, vous pouvez visualiser cette métrique avec CDCLatencyTarget. Si les valeurs CDCLatencySource et CDCLatencyTarget sont toutes les deux élevées, examinez d'abord CDCLatencySource. La valeur CDCSourceLatency peut être égale à 0 lorsqu'il n'y a aucun retard de réplication entre la source et l'instance de réplicati on. CDCSourceLatency peut également atteindre zéro quand la tâche de réplication tente de lire l'événement suivant dans le journal des transactions de la source et qu'il n'y a aucun nouvel événement par rapport à la dernière lecture depuis la source. Lorsque cela se produit, la tâche remet CDCSourceLatency à 0.

Métrique	Description
CDCLatencyTarget	Intervalle, en secondes, entre le premier horodatage d'événement en attente de validation sur la cible et l'horodatage actuel de l'instance AWS DMS. La latence cible est la différence entre l'heure du serveur de l'instance de réplication et le plus ancien identifiant d'événement non confirmé transféré vers un composant cible. En d'autres termes, la latence cible est la différence d'horodatage entre l'instance de réplicati on et le plus ancien événement appliqué mais non confirmé par le point de terminaison TRG (99 %). Quand la valeur CDCLatencyTarget est élevée, cela indique que le processus d'application des événements de modification à la cible est retardé. Pour identifier la latence dans une réplication continue, vous pouvez consulter cette métrique avec CDCLatencySource. Si la valeur CDCLatencyTarget est élevée mais que CDCLatencySource ne l'est pas, vérifiez si : • Aucune clé primaire ni aucun index ne se trouvent dans la cible • Des goulots d'étranglement liés aux ressources se produisent dans la cible ou l'instance de réplication • Des problèmes de réseau résident entre l'instance de réplication et la cible
CPUUtilization	Pourcentage de CPU utilisé par une tâche sur plusieurs cœurs. La sémantique de la valeur CPUUtilization de la tâche est légèremen t différente de celle de la valeur CPUUtilizaiton de la réplication. Si 1 vCPU est entièrement utilisé, cela indique 100 %, mais si plusieurs vCPU sont utilisés, la valeur peut dépasser 100 %. Unités : pourcentage
SwapUsage	Quantité d'échange utilisée par la tâche. Unités : octets

Métrique	Description
MemoryUsage	Groupe de contrôle (cgroup) memory.usage_in_bytes consommé par une tâche. DMS utilise des groupes de contrôle pour contrôler l'utilisation des ressources système telles que la mémoire et le CPU. Cette métrique indique l'utilisation de mémoire d'une tâche en mégaoctets au sein du groupe de contrôle alloué pour cette tâche. Les limites du groupe de contrôle sont basées sur les ressources disponibles pour votre classe d'instances de réplication DMS. memory.usage_in_bytes comprend la taille de résident défini (RSS), le cache et les composants d'échange de mémoire. Le système d'exploitation peut récupérer de la mémoire cache si nécessaire. Nous vous recommandons de surveiller également la métrique de l'instance de réplication, AvailableMemory. AWS DMS augmente cette métrique par rapport aux dimensions combinées de `ReplicationInstanceIdentifer` et `ReplicationTaskIdentifier` dans la console CloudWatch. Utilisez la catégorie `ReplicationInstanceIdentifier, ReplicationTaskIdentifier` pour visualiser cette métrique.

Affichage et gestion des journaux de tâches AWS DMS

Vous pouvez utiliser Amazon CloudWatch pour journaliser les informations sur les tâches au cours d'un processus de migration AWS DMS. Vous activez la journalisation lorsque vous sélectionnez les paramètres de tâche. Pour de plus amples informations, veuillez consulter Paramètres de la tâche de journalisation.

Pour afficher les journaux d'une tâche exécutée, procédez comme suit :

1. Ouvrez la console AWS DMS et choisissez Tâches de migration de base de données dans le panneau de navigation. La boîte de dialogue Tâches de migration de base de données s'affiche.
2. Sélectionnez le nom de votre tâche. La boîte de dialogue Détails de présentation s'affiche.
3. Recherchez la section Journaux de tâche de migration et choisissez Afficher les CloudWatch Logs.

En outre, vous pouvez utiliser AWS CLI ou l'API AWS DMS pour afficher des informations sur les journaux de tâches. Pour ce faire, utilisez la commande AWS CLI `describe-replication-instance-task-logs` ou l'action d'API AWS DMS `DescribeReplicationInstanceTaskLogs`.

Par exemple, la commande AWS CLI suivante indique les métadonnées des journaux de tâches au format JSON.

```
$ aws dms describe-replication-instance-task-logs \
   --replication-instance-arn arn:aws:dms:us-east-1:237565436:rep:CDSFSFSFFFSSUFCAY
```

Voici un exemple de réponse de la commande.

```
{
    "ReplicationInstanceTaskLogs": [
        {
            "ReplicationTaskArn": "arn:aws:dms:us-
east-1:237565436:task:MY34U6Z4MSY52GRTIX3O4AY",
            "ReplicationTaskName": "mysql-to-ddb",
            "ReplicationInstanceTaskLogSize": 3726134
        }
    ],
    "ReplicationInstanceArn": "arn:aws:dms:us-east-1:237565436:rep:CDSFSFSFFFSSUFCAY"
}
```

Dans cette réponse, l'instance de réplication est associée à un seul journal de tâches (`mysql-to-ddb`). La taille de ce journal est de 3 726,124 octets.

Vous pouvez utiliser les informations renvoyés par `describe-replication-instance-task-logs` pour diagnostiquer et résoudre les problèmes liés aux journaux de tâches. Par exemple, si vous autorisez une journalisation détaillée du débogage pour une tâche, le journal de la tâche croîtra rapidement. L'espace de stockage disponible sur l'instance de réplication risque alors d'être entièrement consommé et le statut de l'instance passera à `storage-full`. En décrivant les journaux de tâches, vous pouvez identifier ceux dont vous n'avez plus besoin et les supprimer pour libérer de l'espace de stockage.

Pour supprimer les journaux pour une tâche, définissez le paramètre de tâche `DeleteTaskLogs` sur true. Par exemple, le code JSON suivant supprime les journaux d'une tâche lors de la modification de

la tâche à l'aide de la commande AWS CLI `modify-replication-task` ou de l'action d'API AWS DMS `ModifyReplicationTask`.

```
{
   "Logging": {
       "DeleteTaskLogs":true
   }
}
```

Journalisation des appels d'API AWS DMS avec AWS CloudTrail

AWS DMS est intégré avec ,AWS CloudTrail un service qui fournit un registre des actions prises par un utilisateur, un rôle ou un service AWS dans AWS DMS. CloudTrail capture tous les appels d'API pour AWS DMS en tant qu'événements, y compris les appels émis par la console AWS DMS et les appels de code transmis aux opérations d'API AWS DMS. Si vous créez un journal d'activité, vous pouvez activer la livraison continue d'événements CloudTrail à un compartiment Amazon S3, y compris des événements pour AWS DMS. Si vous ne configurez pas de journal de suivi, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Event history (Historique des événements). Avec les informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée à l'AWS DMS, ainsi que l'adresse IP, l'auteur et date de la demande, ainsi que d'autres détails.

Pour en savoir plus sur CloudTrail, consultez le Guide de l'utilisateur AWS CloudTrail.

AWS DMS Informations dans CloudTrail

CloudTrail est activé dans votre compte AWS lors de sa création. Lorsqu'une activité a lieu dans AWS DMS, cette activité est enregistrée dans un événement CloudTrail avec d'autres AWS événements de service dans Historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS. Pour de plus amples informations, veuillez consulter Affichage des événements avec l'historique des événements CloudTrail.

Pour enregistrer en continu les événements dans votre compte AWS, y compris les événements d'AWS DMS, créez un journal d'activité. Un journal d'activité permet à CloudTrail de distribuer les fichiers journaux vers Amazon S3 bucket. Par défaut, lorsque vous créez un journal de suivi dans la console, il s'applique à toutes les régions AWS. Le journal d'activité consigne les événements de

toutes les Régions dans la partition AWSAWS et transfère les fichiers journaux dans le compartiment Amazon S3 de votre choix. En outre, vous pouvez configurer d'autres services AWS pour analyser plus en profondeur les données d'événement collectées dans les journaux CloudTrail et agir sur celles-ci. Pour plus d'informations, consultez :

- Présentation de la création d'un journal d'activité
- Intégrations et services pris en charge par CloudTrail
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers journaux CloudTrail de plusieurs régions AWS et Réception de fichiers journaux CloudTrail de plusieurs comptes

Toutes les actions AWS DMS sont consignées par CloudTrail et documentées dans la Référence des API AWS Database Migration Service. À titre d'exemple, les appels vers les actions `CreateReplicationInstance`, `TestConnection` et `StartReplicationTask` génèrent des entrées dans les fichiers journaux CloudTrail.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les autorisations utilisateur root ou IAM .
- Si la demande a été effectuée avec des autorisations de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour plus d'informations, consultez l'élément userIdentity CloudTrail.

Présentation des AWS DMS entrées des fichiers journaux

Un journal d'activité est une configuration qui permet d'envoyer les événements dans des fichiers journaux à un compartiment Amazon S3 que vous spécifiez. Les fichiers journaux CloudTrail peuvent contenir une ou plusieurs entrées. Un événement représente une demande individuelle émise à partir d'une source quelconque et comprend des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. Les fichiers journaux CloudTrail ne constituent pas une trace de pile ordonnée d'appels d'API publics. Ils ne suivent donc aucun ordre précis.

L'exemple suivant montre une entrée de journal CloudTrail qui illustre l'action `RebootReplicationInstance`.

```
    {
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE:johndoe",
        "arn": "arn:aws:sts::123456789012:assumed-role/admin/johndoe",
        "accountId": "123456789012",
        "accessKeyId": "ASIAYFI33SINADOJJEZW",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-08-01T16:42:09Z"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/admin",
                "accountId": "123456789012",
                "userName": "admin"
            }
        }
    },
    "eventTime": "2018-08-02T00:11:44Z",
    "eventSource": "dms.amazonaws.com",
    "eventName": "RebootReplicationInstance",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "72.21.198.64",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "forceFailover": false,
        "replicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:EX4MBJ2NMRDL3BMAYJOXUGYPUE"
    },
    "responseElements": {
        "replicationInstance": {
            "replicationInstanceIdentifier": "replication-instance-1",
            "replicationInstanceStatus": "rebooting",
            "allocatedStorage": 50,
            "replicationInstancePrivateIpAddresses": [
                "172.31.20.204"
            ],
            "instanceCreateTime": "Aug 1, 2018 11:56:21 PM",
```

```
            "autoMinorVersionUpgrade": true,
            "engineVersion": "2.4.3",
            "publiclyAccessible": true,
            "replicationInstanceClass": "dms.t2.medium",
            "availabilityZone": "us-east-1b",
            "kmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
f7bc0f8e-1a3a-4ace-9faa-e8494fa3921a",
            "replicationSubnetGroup": {
                "vpcId": "vpc-1f6a9c6a",
                "subnetGroupStatus": "Complete",
                "replicationSubnetGroupArn": "arn:aws:dms:us-
east-1:123456789012:subgrp:EDHRVRBAAAPONQAIYWP4NUW22M",
                "subnets": [
                    {
                        "subnetIdentifier": "subnet-cbfff283",
                        "subnetAvailabilityZone": {
                            "name": "us-east-1b"
                        },
                        "subnetStatus": "Active"
                    },
                    {
                        "subnetIdentifier": "subnet-d7c825e8",
                        "subnetAvailabilityZone": {
                            "name": "us-east-1e"
                        },
                        "subnetStatus": "Active"
                    },
                    {
                        "subnetIdentifier": "subnet-6746046b",
                        "subnetAvailabilityZone": {
                            "name": "us-east-1f"
                        },
                        "subnetStatus": "Active"
                    },
                    {
                        "subnetIdentifier": "subnet-bac383e0",
                        "subnetAvailabilityZone": {
                            "name": "us-east-1c"
                        },
                        "subnetStatus": "Active"
                    },
                    {
                        "subnetIdentifier": "subnet-42599426",
                        "subnetAvailabilityZone": {
```

```
                            "name": "us-east-1d"
                        },
                        "subnetStatus": "Active"
                    },
                    {
                        "subnetIdentifier": "subnet-da327bf6",
                        "subnetAvailabilityZone": {
                            "name": "us-east-1a"
                        },
                        "subnetStatus": "Active"
                    }
                ],
                "replicationSubnetGroupIdentifier": "default-vpc-1f6a9c6a",
                "replicationSubnetGroupDescription": "default group created by console
 for vpc id vpc-1f6a9c6a"
            },
            "replicationInstanceEniId": "eni-0d6db8c7137cb9844",
            "vpcSecurityGroups": [
                {
                    "vpcSecurityGroupId": "sg-f839b688",
                    "status": "active"
                }
            ],
            "pendingModifiedValues": {},
            "replicationInstancePublicIpAddresses": [
                "18.211.48.119"
            ],
            "replicationInstancePublicIpAddress": "18.211.48.119",
            "preferredMaintenanceWindow": "fri:22:44-fri:23:14",
            "replicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:EX4MBJ2NMRDL3BMAYJOXUGYPUE",
            "replicationInstanceEniIds": [
                "eni-0d6db8c7137cb9844"
            ],
            "multiAZ": false,
            "replicationInstancePrivateIpAddress": "172.31.20.204",
            "patchingPrecedence": 0
        }
    },
    "requestID": "a3c83c11-95e8-11e8-9d08-4b8f2b45bfd5",
    "eventID": "b3c4adb1-e34b-4744-bdeb-35528062a541",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
```

```
}
```

Journalisation du contexte AWS DMS

AWS DMS utilise la journalisation du contexte pour vous fournir des informations sur une migration en cours. La journalisation du contexte écrit des informations, telles que les suivantes, dans le journal CloudWatch de la tâche :

- Informations sur la connexion de la tâche aux bases de données source et cible.
- Comportement des tâches de réplication. Vous pouvez utiliser les journaux de tâches pour diagnostiquer les problèmes de réplication.
- Instructions SQL sans données qu'AWS DMS exécute sur les bases de données source et cible. Vous pouvez utiliser les journaux SQL pour diagnostiquer un comportement de migration inattendu.
- Détails de position de flux pour chaque événement CDC.

La journalisation du contexte n'est disponible que dans AWS DMS version 3.5.0 ou ultérieure.

AWS DMS active par défaut la journalisation du contexte. Pour contrôler la journalisation du contexte, définissez le paramètre de tâche `EnableLogContext` sur `true` ou `false`, ou modifiez la tâche dans la console.

AWS DMS écrit les informations de journalisation du contexte dans la tâche de réplication du journal CloudWatch toutes les trois minutes. Veillez à ce que votre instance de réplication dispose de suffisamment d'espace pour son journal d'application. Pour plus d'informations sur la gestion des journaux de tâches, consultez Affichage et gestion des journaux de tâches AWS DMS.

Rubriques

- Types d'objets
- Exemples de journalisation
- Limites

Types d'objets

AWS DMS produit une journalisation du contexte dans CloudWatch pour les types d'objets suivants.

Type d'objet	Description
TABLE_NAME	Ces entrées de journal contiennent des informations sur les tables concernées par la règle de mappage des tâches actuelle. Vous pouvez utiliser ces entrées pour examiner les événements de table pour une période spécifique au cours de la migration.
SCHEMA_NAME	Ces entrées de journal contiennent des informations sur les schémas utilisés par la règle de mappage des tâches actuelle. Vous pouvez utiliser ces entrées pour déterminer le schéma qu'AWS DMS utilise pendant une période spécifique au cours de la migration.
TRANSACTION_ID	Ces entrées contiennent l'ID de transaction pour chaque modification DML/DDL capturée à partir de la base de données source. Vous pouvez utiliser ces entrées de journal pour déterminer les changements survenus au cours d'une transaction donnée.
CONNECTION_ID	Ces entrées contiennent l'ID de connexion. Vous pouvez utiliser ces entrées de journal pour déterminer la connexion qu'AWS DMS utilise pour chaque étape de migration.
STATEMENT	Ces entrées contiennent le code SQL utilisé pour récupérer, traiter et appliquer chaque modification de migration.
STREAM_POSITION	Ces entrées contiennent la position dans le fichier journal de transactions pour chaque action de migration sur la base de données source. Le format de ces entrées varie selon le type de moteur de base de données source. Vous pouvez également utiliser ces informati

Type d'objet	Description
	ons pour déterminer une position de départ d'un point de contrôle de récupération lors de la configuration de la réplication CDC uniquemen t.

Exemples de journalisation

Cette section contient des exemples d'enregistrements de journal que vous pouvez utiliser pour surveiller la réplication et diagnostiquer les problèmes de réplication.

Exemples de journaux de connexion

Cette section contient des exemples de journaux qui incluent les ID de connexion.

```
2023-02-22T10:09:29 [SOURCE_CAPTURE  ]I:  Capture record 1 to internal
 queue from Source  {operation:START_REGULAR (43), connectionId:27598,
 streamPosition:0000124A/6800A778.NOW}  (streamcomponent.c:2920)

2023-02-22T10:12:30 [SOURCE_CAPTURE  ]I:  Capture record 0 to internal queue from
 Source  {operation:IDLE (51), connectionId:27598}  (streamcomponent.c:2920)

2023-02-22T11:25:27 [SOURCE_CAPTURE  ]I:  Capture record 0 to internal queue
 from Source  {operation:IDLE (51), columnName:region, connectionId:27598}
 (streamcomponent.c:2920)
```

Exemples de journaux de comportement des tâches

Cette section contient des exemples de journaux relatifs au comportement des tâches de réplication. Vous pouvez utiliser ces informations pour diagnostiquer les problèmes de réplication, tels que le statut `IDLE` d'une tâche.

Les journaux `SOURCE_CAPTURE` suivants indiquent qu'aucun événement ne peut être lu dans le fichier journal de la base de données source et contiennent des enregistrements `TARGET_APPLY` indiquant qu'aucun événement reçu des composants AWS DMS CDC ne s'applique à la base de données cible. Ces événements contiennent également des détails contextuels relatifs à des événements précédemment appliqués.

```
2023-02-22T11:23:24 [SOURCE_CAPTURE  ]I:  No Event fetched from wal log
 (postgres_endpoint_wal_engine.c:1369)
2023-02-22T11:24:29 [TARGET_APPLY    ]I:  No records received to load
 or apply on target , waiting for data from upstream. The last context
 is  {operation:INSERT (1), tableName:sales_11, schemaName:public,
 txnId:18662441, connectionId:17855, statement:INSERT INTO
 "public"."sales_11"("sales_no","dept_name","sale_amount","sale_date","region") values
 (?,?,?,?,?),
```

Exemples de journaux d'instructions SQL

Cette section contient des exemples de journaux relatifs aux instructions SQL exécutées sur les bases de données source et cible. Les instructions SQL que vous voyez dans ces journaux indiquent uniquement l'instruction SQL et ne montrent pas les données. Le journal `TARGET_APPLY` suivant montre une instruction `INSERT` exécutée sur la cible.

```
2023-02-22T11:26:07 [TARGET_APPLY    ]I:  Applied record 2193305 to
 target  {operation:INSERT (1), tableName:sales_111, schemaName:public,
 txnId:18761543, connectionId:17855, statement:INSERT INTO
 "public"."sales_111"("sales_no","dept_name","sale_amount","sale_date","region") values
 (?,?,?,?,?),
```

Limites

Les limitations suivantes s'appliquent à la journalisation du contexte par AWS DMS :

- AWS DMS crée une journalisation minimale de tous les types de point de terminaison, alors qu'une journalisation de contexte extensive, spécifique au moteur, est disponible uniquement pour les types de point de terminaison suivants. Nous vous recommandons d'activer la journalisation du contexte lorsque vous utilisez ces types de point de terminaison.
  - MySQL
  - PostgreSQL
  - Oracle
  - Microsoft SQL Server
  - MongoDB / Amazon DocumentDB
  - Amazon S3

Utilisation des événements et des notifications Amazon EventBridge dans AWS Database Migration Service

Vous pouvez utiliser Amazon EventBridge pour fournir une notification quand un événement AWS DMS se produit, tel que la création ou la suppression d'une instance de réplication. EventBridge reçoit les événements et route la notification d'un événement conformément aux règles d'événement. Vous pouvez utiliser les notifications sous n'importe quelle forme prise en charge par Amazon EventBridge pour une région AWS. Pour plus d'informations sur Amazon EventBridge, consultez Qu'est-ce qu'Amazon EventBridge ? dans le Guide de l'utilisateur Amazon EventBridge.

> **Note**
>
> L'utilisation des événements Amazon EventBridge est prise en charge dans AWS DMS versions 3.4.5 et ultérieures.

EventBridge reçoit un événement, un indicateur d'un changement dans l'environnement AWS DMS, et applique une règle pour router cet événement vers un mécanisme de notification. Les règles font correspondre les événements aux mécanismes de notification en fonction de la structure de l'événement, appelée modèle d'événement.

AWS DMS groupe les événements en catégories auxquelles vous pouvez appliquer une règle d'événement, afin d'être averti lorsqu'un événement de cette catégorie se produit. Supposons, par exemple, que vous appliquiez une règle d'événement EventBridge à la catégorie Création pour une instance de réplication donnée. Vous êtes ensuite averti chaque fois qu'un événement lié à une création survient qui affecte votre instance de réplication. Si vous appliquez une règle à la catégorie Modification de configuration pour une instance de réplication, vous recevez une notification quand la configuration de l'instance de réplication est modifiée. Pour obtenir la liste des catégories d'événements fournies par AWS DMS, consultez les catégories d'événements et les messages d'événements AWS DMS ci-dessous.

> **Note**
>
> Pour autoriser la publication à partir de events.amazonaws.com, veillez à mettre à jour les stratégies d'accès de vos rubriques Amazon SNS. Pour plus d'informations, consultez Utilisation de politiques basées sur les ressources pour Amazon EventBridge dans le Guide de l'utilisateur Amazon EventBridge.

Pour plus d'informations sur le transfert des abonnements aux événements vers Amazon EventBridge, consultez Migration des abonnements aux événements actifs de DMS vers Amazon EventBridge.
Pour plus d'informations sur l'utilisation des messages texte avec Amazon SNS, consultez Envoi et réception de notifications par SMS avec Amazon SNS.

Utilisation des règles d'événement Amazon EventBridge pour AWS DMS

Amazon EventBridge envoie des notifications d'événement aux adresses que vous fournissez lorsque vous créez une règle d'événement EventBridge. Vous pouvez créer plusieurs règles différentes. Par exemple, vous pouvez créer une règle recevant toutes les notifications d'événement et une autre règle incluant uniquement des évènements critiques pour vos ressources DMS de production. Vous pouvez également activer ou désactiver les notifications d'événement dans EventBridge.

Pour créer des règles Amazon EventBridge qui réagissent aux événements AWS DMS

- Suivez les étapes décrites dans Création de règles Amazon EventBridge qui réagissent aux événements dans le Guide de l'utilisateur Amazon EventBridge, et créez une règle pour les événements AWS DMS :

  a. Spécifiez une action de notification à effectuer quand EventBridge reçoit un événement correspondant au modèle d'événement figurant dans la règle. Quand un événement correspond, EventBridge envoie cet événement et invoque l'action définie dans la règle.
  b. Pour Service Provider (Fournisseur de service), sélectionnez AWS.
  c. Pour Nom du service, choisissez Database Migration Service (DMS).

Vous pouvez alors commencer à recevoir des notifications d'événement.

L'exemple JSON suivant montre un modèle d'événement EventBridge pour un service AWS DMS.

```
{
   "version":"0",
   "id":"11a11b11-222b-333a-44d4-01234a5b67890",
   "detail-type":"DMS Replication Task State Change",
   "source":"aws.dms",
   "account":"0123456789012",
```

```
   "time":"1970-01-01T00:00:00Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:dms:us-east-1:012345678901:task:AAAABBBB0CCCCDDDDEEEEE1FFFF2GGG3FFFFFF3"
   ],
   "detail":{
      "type":"REPLICATION_TASK",
      "category":"StateChange",
      "eventType":"REPLICATION_TASK_STARTED",
      "eventId":"DMS-EVENT-0069",
      "resourceLink":"https://console.aws.amazon.com/dms/v2/home?region=us-
east-1#taskDetails/taskName",
      "detailMessage":"Replication task started, with flag = fresh start"
   }
}
```

Pour obtenir la liste des catégories et événements dont vous pouvez être informé, consultez la section suivante.

Catégories d'événements et messages d'événements AWS DMS

AWS DMS génère un nombre important d'événements dans des catégories que vous pouvez identifier. Chaque catégorie s'applique à un type de source d'instance de réplication ou de tâche de réplication.

Rubriques

- Messages d'événements ReplicationInstance
- Messages d'événements ReplicationTask
- Messages d'événements Replication

Messages d'événements ReplicationInstance

Le tableau suivant recense les catégories et événements possibles pour le type de source ReplicationInstance.

Catégorie	ID d'événement	Description
Création	DMS-EVENT-0067	Une instance de réplication est en cours de création.
Suppression	DMS-EVENT-0066	L'instance de réplication est en cours de suppression.
Modification de configuration	DMS-EVENT-0012	La classe d'instances de réplication de cette instance de réplication est en cours de modification.
Modification de configuration	DMS-EVENT-0018	L'espace de stockage de l'instance de réplication est en cours d'augmentation.
Modification de configuration	DMS-EVENT-0024	L'instance de réplication est en cours de transition vers une configuration multi-AZ.
Modification de configuration	DMS-EVENT-0030	L'instance de réplication est en cours de transition vers une configuration mono-AZ.
Maintenance	DMS-EVENT-0026	La maintenance hors connexion de l'instance de réplication est en cours. L'instance de réplication n'est pas disponible actuellement.
Création	DMS-EVENT-0005	Une instance de réplication a été créée.
Suppression	DMS-EVENT-0003	L'instance de réplication a été supprimée.
Modification de configuration	DMS-EVENT-0014	La classe d'instances de réplication de cette instance de réplication a changé.

Catégorie	ID d'événement	Description
Modification de configuration	DMS-EVENT-0017	L'espace de stockage de l'instance de réplication a été augmenté.
Modification de configuration	DMS-EVENT-0025	L'instance de réplication a terminé sa transition vers une configuration multi-AZ.
Modification de configuration	DMS-EVENT-0029	L'instance de réplication a terminé sa transition vers une configuration mono-AZ.
Maintenance	DMS-EVENT-0047	Le logiciel de gestion figurant sur l'instance de réplication a été mis à jour.
Maintenance	DMS-EVENT-0027	La maintenance hors connexion de l'instance de réplication est terminée. L'instance de réplication est désormais disponible.
Maintenance	DMS-EVENT-0068	L'instance de réplication est dans un état qui ne peut pas être mis à niveau.
Basculement	DMS-EVENT-0034	Si vous demandez trop souvent un basculement, cet événement se produit à la place des événements de basculement standard.
Échec	DMS-EVENT-0031	L'instance de réplication a été placée dans l'état %s.
Échec	DMS-EVENT-0036	L'instance de réplication a échoué en raison d'une incompatibilité de réseau.

Catégorie	ID d'événement	Description
Échec	DMS-EVENT-0037	Quand le service ne peut pas accéder à la clé KMS utilisée pour chiffrer le volume de données.
Échec		L'instance de réplication a été placée dans l'état incompatible-parameters.
Basculement		Le délai d'attente pour qu'un état sécurisé initie le basculement demandé par l'utilisateur a expiré.
Basculement	DMS-EVENT-0013	Le basculement a commencé pour une instance de réplication multi-AZ.
Basculement	DMS-EVENT-0049	Le basculement est terminé pour une instance de réplication multi-AZ.
Basculement	DMS-EVENT-0050	L'activation multi-AZ a commencé.
Basculement	DMS-EVENT-0051	L'activation multi-AZ est terminée.
Changement d'état		Les journaux des requêtes générales et lentes ont fait l'objet d'une rotation automatique en tant que %s.

Catégorie	ID d'événement	Description
Changement d'état		AWS DMS ne parvient pas à accéder à la clé de chiffrement KMS pour l'instance d'application %s. Cela est probablement dû à la désactivation de la clé ou à l'impossibilité pour AWS DMS d'y accéder. Si cela continue, l'application sera placée dans un état inaccessible. Veuillez vous reporter à la section sur la résolution des problèmes dans la documentation AWS DMS pour plus de détails.
Changement d'état		AWS DMS parvient désormais à accéder à la clé de chiffrement KMS pour l'instance d'application %s.
Changement d'état		Amazon DMS ne parvient pas à accéder à la clé de chiffrement KMS pour l'instance d'application %s. Cette application sera placée dans un état inaccessible. Veuillez vous reporter à la section sur la résolution des problèmes dans la documentation Amazon DMS pour plus de détails.
Changement d'état		Redémarrage de l'application sur HM dans le cadre de la création d'une instance de réplication
Changement d'état		Arrêt de l'application sur HM dans le cadre de la suppression de l'instance de réplication
Basculement	DMS-EVENT-0015	Basculement multi-AZ vers l'instance de secours terminé.

Catégorie	ID d'événement	Description
Stockage faible	DMS-EVENT-0007	L'espace de stockage disponible pour l'instance de réplication est faible.
Stockage faible		Les inodes alloués sont épuisés : mettre à l'échelle le stockage pour résoudre le problème

Messages d'événements ReplicationTask

Le tableau suivant recense les catégories et événements possibles pour le type de source ReplicationTask.

Catégorie	ID d'événement	Description
Échec	DMS-EVENT-0078	Une tâche de réplication a échoué.
Échec	DMS-EVENT-0082	Un appel pour nettoyer les données d'une tâche a échoué.
Modification d'état	DMS-EVENT-0081	Le rechargement des détails d'une table a été demandé.
Modification d'état		La tâche de réplication a été copiée.
Modification d'état		La copie de la tâche de réplication a échoué.
Modification d'état		La tâche de réplication a été déplacée.
Modification d'état		Le déplacement de la tâche de réplicati on a échoué.
Modification d'état		La création de la tâche cible a échoué.

Catégorie	ID d'événement	Description
Modification d'état		L'exécution d'évaluation de la tâche de réplication a commencé.
Modification d'état		L'exécution d'évaluation de la tâche de réplication s'est terminée avec succès.
Modification d'état		L'exécution d'évaluation de la tâche de réplication s'est terminée sur un échec.
Changement d'état		L'exécution d'évaluation de la tâche de réplication s'est terminée avec un avertissement.
Changement d'état		L'exécution d'évaluation de la tâche de réplication s'est terminée avec une erreur.
Changement d'état		L'exécution %s d'évaluation de la tâche de réplication a été annulée.
Changement d'état		L'exécution %s d'évaluation de la tâche de réplication a été supprimée.
Changement d'état		L'exécution d'évaluation de la tâche de réplication n'a pas pu provisionner les ressources.
Changement d'état		La tâche de réplication a échoué.
Création		La tâche de réplication a été créée.
Changement de configuration		Une tâche de réplication a été modifiée.

Catégorie	ID d'événement	Description
Échec		Une tâche de réplication a échoué.
Changement d'état	DMS-EVENT-0091	Lecture mise en pause, limite de fichiers d'échange atteinte.
Changement d'état	DMS-EVENT-0092	Lecture mise en pause, limite d'utilisation du disque atteinte.
Changement d'état	DMS-EVENT-0093	Lecture mise en pause, limite d'utilisation du disque atteinte.
Changement d'état	DMS-EVENT-0093	La lecture a repris.
Changement d'état	DMS-EVENT-0069	La tâche de réplication a démarré avec taskType: %s, startType: %s
Changement d'état	DMS-EVENT-0079	La tâche de réplication s'est arrêtée.
Suppression	DMS-EVENT-0073	La tâche de réplication a été supprimée .

Messages d'événements Replication

Le tableau suivant recense les catégories et événements possibles pour le type de source Replication.

Catégorie	Description
Modification d'état	Événement d'augmentation de la réplication DMS.
Modification d'état	Événement de réduction de la réplication DMS.
Modification d'état	L'événement de mise à l'échelle de la réplication DMS est terminé.
Modification d'état	La réplication DMS a été créée.

Catégorie	Description
Modification d'état	La réplication DMS est en cours d'initialisation.
Modification d'état	La réplication DMS prépare les ressources requises pour la collecte des métadonnées.
Modification d'état	Les connexions liées à la réplication DMS sont actuellement testées.
Modification d'état	La réplication DMS récupère les métadonnées.
Modification d'état	La réplication DMS calcule la capacité.
Modification d'état	La réplication DMS provisionne sa capacité.
Modification d'état	La réplication DMS a été provisionnée.
Modification d'état	La réplication DMS a commencé.
Modification d'état	La réplication DMS est en cours d'exécution.
Modification d'état	La réplication DMS est en cours d'arrêt.
Modification d'état	La réplication DMS s'est arrêtée.
Modification d'état	La réplication DMS est en cours de modification.
Modification d'état	La réplication DMS est en cours de suppression.
Modification d'état	La réplication DMS déprovisionne sa capacité.
Modification d'état	La réplication DMS a fait l'objet d'un déprovisionnement.
Échec	La réplication DMS a échoué.

Utilisation des événements et des notifications Amazon SNS dans AWS Database Migration Service

Depuis la publication d'AWS DMS 3.4.5 et avec les versions ultérieures, nous vous recommandons d'utiliser Amazon EventBridge pour envoyer des notifications quand un événement AWS DMS se produit. Pour plus d'informations sur l'utilisation des événements EventBridge avec AWS DMS, consultez Utilisation des événements et des notifications Amazon EventBridge dans AWS Database Migration Service.

Transfert des abonnements aux événements vers Amazon EventBridge

Vous pouvez utiliser la commande AWS CLI suivante pour migrer les abonnements aux événements actifs à partir de DMS vers Amazon EventBridge, jusqu'à 10 à la fois.

```
update-subscriptions-to-event-bridge [--force-move | --no-force-move]
```

Par défaut, AWS DMS migre uniquement les abonnements aux événements actifs lorsque votre instance de réplication est à jour avec AWS DMS 3.4.5 ou version ultérieure. Pour remplacer ce comportement par défaut, utilisez l'option `--force-move`. Toutefois, certains types d'événements peuvent ne pas être disponibles en utilisant Amazon EventBridge si vos instances de réplication ne sont pas mises à niveau.

Pour exécuter la commande CLI `update-subscriptions-to-event-bridge`, un utilisateur AWS Identity and Access Management (IAM) doit disposer des autorisations de politique suivantes.

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
          "Effect": "Allow",
          "Action": [
              "SNS:GetTopicAttributes",
              "SNS:SetTopicAttributes",
              "events:PutTargets",
              "events:EnableRule",
              "events:PutRule"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

Pour plus d'informations sur le transfert des abonnements vers EventBridge, consultez UpdateSubscriptionsToEventBridge dans Référence d'API AWS Database Migration Service.

Utilisation des événements et des notifications Amazon SNS

AWS DMS versions 3.4.5 et antérieures prennent en charge le traitement des événements et des notifications comme décrit ci-dessous.

AWS Database Migration Service (AWS DMS) peut utiliser Amazon Simple Notification Service (Amazon SNS) pour envoyer des notifications lorsqu'un événement AWS DMS se produit, par exemple lors de la création ou de la suppression d'une instance de réplication. Vous pouvez utiliser ces notifications sous n'importe quelle forme prise en charge par Amazon SNS pour une région AWS, telle qu'un e-mail, un message texte ou un appel à un point de terminaison HTTP.

AWS DMS groupe les événements en catégories auxquelles vous pouvez vous abonner afin d'être averti lorsqu'un événement de cette catégorie se produit. Par exemple, si vous vous abonnez à la catégorie de création d'une instance de réplication donnée, vous recevez une notification chaque fois que survient un événement lié à la création qui affecte votre instance de réplication. Si vous vous abonnez à la catégorie Modification de configuration pour une instance de réplication, vous recevez une notification en cas de modification de la configuration de l'instance de réplication. Vous recevez également une notification en cas de modification d'un abonnement à une notification d'évènements. Pour obtenir la liste des catégories d'événements fournies par AWS DMS, consultez Catégories d'événements et messages d'événements AWS DMS pour les notifications SNS, ci-dessous.

AWS DMS envoie des notifications d'événement aux adresses que vous fournissez lorsque vous créez un abonnement aux événements. Il se peut que vous souhaitiez créer plusieurs abonnements différents, tel qu'un abonnement recevant toutes les notifications d'événements et un autre abonnement incluant uniquement les événements critiques pour vos ressources DMS de production. Vous pouvez facilement désactiver les notifications sans supprimer un abonnement. Pour ce faire, désélectionnez l'option Activé dans la console AWS DMS ou définissez le paramètre `Enabled` sur false à l'aide de l'API AWS DMS.

> **Note**
>
> Les notifications d'événement AWS DMS utilisant les messages texte SMS sont actuellement disponibles pour les ressources AWS DMS dans toutes les régions AWS où Amazon SNS est pris en charge. Pour obtenir la liste des pays et des régions AWS où Amazon SNS prend en charge la messagerie SMS, consultez Pays et régions pris en charge.
> Pour plus d'informations sur l'utilisation des SMS avec SNS, consultez Envoi et réception de notifications par SMS avec Amazon SNS.
> Les notifications d'événement AWS DMS diffèrent des événements CloudTrail dans CloudWatch ou EventBridge. Les notifications d'événement CloudTrail peuvent être générées par n'importe quelle invocation d'API. DMS envoie une notification uniquement quand un événement DMS se produit.

AWS DMS utilise un identificateur d'abonnement pour identifier chaque abonnement. Plusieurs abonnements aux événements AWS DMS peuvent être publiés dans la même rubrique Amazon SNS. Lorsque vous utilisez la notification d'événement, les tarifs Amazon SNS s'appliquent ; pour plus d'informations sur la facturation d'Amazon SNS, consultez Tarification d'Amazon SNS.

Pour vous abonner à des événements AWS DMS avec Amazon SNS, utilisez le processus suivant :

1. Créer une rubrique Amazon SNS. Dans la rubrique, indiquez le type de notification que vous souhaitez recevoir et à quelle adresse ou quel numéro elle doit être envoyée.
2. Créez un abonnement aux notifications d'événement AWS DMS à l'aide de la AWS Management Console, d'AWS CLI ou de l'API AWS DMS.
3. AWS DMS envoie un e-mail d'approbation ou un SMS aux adresses que vous avez fournies avec votre abonnement. Pour valider votre abonnement, cliquez sur le lien dans l'e-mail d'approbation ou le SMS.
4. Lorsque vous avez confirmé l'abonnement, le statut de celui-ci est mis à jour dans la section Abonnements aux événements de la console AWS DMS.
5. Vous commencez alors à recevoir des notifications d'évènements.

Pour obtenir la liste des catégories et événements dont vous pouvez être informé, consultez la section suivante. Pour plus de détails sur l'abonnement et l'utilisation des abonnements aux événements AWS DMS, consultez Abonnement aux notifications d'événement AWS DMS à l'aide de SNS.

Catégories d'événements et messages d'événements AWS DMS pour les notifications SNS

> **Important**
>
> Depuis la publication d'AWS DMS 3.4.5 et avec les versions ultérieures, nous vous recommandons d'utiliser Amazon EventBridge pour envoyer des notifications quand un événement AWS DMS se produit. Pour plus d'informations sur l'utilisation des événements EventBridge avec AWS DMS, consultez Utilisation des événements et des notifications Amazon EventBridge dans AWS Database Migration Service.

AWS DMS génère un nombre important d'événements dans des catégories auxquelles vous pouvez vous abonner à l'aide de la console AWS DMS ou de l'API AWS DMS. Chaque catégorie s'applique à un type de source ; AWS DMS prend actuellement en charge les types de source suivants : instance de réplication et tâche de réplication.

Le tableau suivant recense les catégories et événements possibles pour le type de source instance de réplication.

Catégorie	ID d'événement DMS	Description
Modification de configuration	DMS-EVENT-0012	La classe d'instances de réplication de cette instance de réplication est en cours de modificat ion.
Modification de configuration	DMS-EVENT-0014	La classe d'instances de réplication de cette instance de réplication a changé.
Modification de configuration	DMS-EVENT-0018	L'espace de stockage de l'instance de réplication est en cours d'augmentation.
Modification de configuration	DMS-EVENT-0017	L'espace de stockage de l'instance de réplication a été augmenté.
Modification de configuration	DMS-EVENT-0024	L'instance de réplication est en cours de transition vers une configuration multi-AZ.

Catégorie	ID d'événement DMS	Description
Modification de configuration	DMS-EVENT-0025	L'instance de réplication a terminé sa transition vers une configuration multi-AZ.
Modification de configuration	DMS-EVENT-0030	L'instance de réplication est en cours de transition vers une configuration mono-AZ.
Modification de configuration	DMS-EVENT-0029	L'instance de réplication a terminé sa transition vers une configuration mono-AZ.
Création	DMS-EVENT-0067	Une instance de réplication est en cours de création.
Création	DMS-EVENT-0005	Une instance de réplication est créée.
Suppression	DMS-EVENT-0066	L'instance de réplication est en cours de suppressi on.
Suppression	DMS-EVENT-0003	L'instance de réplication est supprimée.
Maintenance	DMS-EVENT-0047	Le logiciel de gestion figurant sur l'instance de réplication a été mis à jour.
Maintenance	DMS-EVENT-0026	La maintenance hors connexion de l'instance de réplication est en cours. L'instance de réplication n'est pas disponible actuellement.
Maintenance	DMS-EVENT-0027	La maintenance hors connexion de l'instance de réplication est terminée. L'instance de réplication est désormais disponible.
Maintenance	DMS-EVENT-0068	Une instance de réplication est dans un état qui ne peut pas être mis à niveau.

Catégorie	ID d'événement DMS	Description
Stockage faible	DMS-EVENT-0007	L'instance de réplication a consommé plus de 90 % de son stockage alloué. Vous pouvez surveiller l'espace de stockage d'une instance de réplicati on à l'aide de la métrique Espace de stockage disponible.
Basculement	DMS-EVENT-0013	Le basculement a commencé pour une instance de réplication multi-AZ.
Basculement	DMS-EVENT-0049	Le basculement est terminé pour une instance de réplication multi-AZ.
Basculement	DMS-EVENT-0015	Le basculement multi-AZ vers l'instance de secours est terminé.
Basculement	DMS-EVENT-0050	L'activation multi-AZ a commencé.
Basculement	DMS-EVENT-0051	L'activation multi-AZ est terminée.
Basculement	DMS-EVENT-0034	Si vous demandez trop souvent un basculeme nt, cet événement se produit à la place des événements de basculement standard.
Échec	DMS-EVENT-0031	L'instance de réplication a subi un échec de stockage.
Échec	DMS-EVENT-0036	L'instance de réplication a échoué en raison d'une incompatibilité de réseau.
Échec	DMS-EVENT-0037	Le service ne peut pas accéder à la clé AWS KMS utilisée pour chiffrer le volume de données.

Le tableau suivant recense les catégories et événements possibles pour le type de source tâche de réplication.

Catégorie	ID d'événement DMS	Description
Modification d'état	DMS-EVENT-0069	La tâche de réplication a commencé.
Modification d'état	DMS-EVENT-0081	Un rechargement des détails d'une table a été demandé.
Modification d'état	DMS-EVENT-0079	La tâche de réplication s'est arrêtée.
Modification d'état	DMS-EVENT-0091	Lecture mise en pause, limite de fichiers d'échange atteinte.
Modification d'état	DMS-EVENT-0092	Lecture mise en pause, limite d'utilisation du disque atteinte.
Modification d'état	DMS-EVENT-0093	La lecture a repris.
Échec	DMS-EVENT-0078	La tâche de réplication a échoué.
Échec	DMS-EVENT-0082	Un appel de suppression de la tâche n'a pas réussi à nettoyer les données de la tâche.
Modification de configuration	DMS-EVENT-0080	La tâche de réplication est modifiée.
Suppression	DMS-EVENT-0073	La tâche de réplication est supprimée.
Création	DMS-EVENT-0074	La tâche de réplication est créée.

L'exemple suivant montre un abonnement aux événements AWS DMS de la catégorie Modification d'état.

```
            Resources:
                DMSEvent:
                    Type: AWS::DMS::EventSubscription
```

```
                Properties:
                    Enabled: true
                    EventCategories: State Change
                    SnsTopicArn: arn:aws:sns:us-east-1:123456789:testSNS
                    SourceIds: []
                    SourceType: replication-task
```

Abonnement aux notifications d'événement AWS DMS à l'aide de SNS

> **Important**
>
> Depuis la publication d'AWS DMS 3.4.5 et avec les versions ultérieures, nous vous recommandons d'utiliser Amazon EventBridge pour envoyer des notifications quand un événement AWS DMS se produit. Pour plus d'informations sur l'utilisation des événements EventBridge avec AWS DMS, consultez Utilisation des événements et des notifications Amazon EventBridge dans AWS Database Migration Service.

Vous pouvez créer un abonnement aux notifications d'événements AWS DMS afin de pouvoir être informé quand un événement AWS DMS se produit. La solution la plus simple pour créer un abonnement consiste à utiliser la console AWS DMS. Dans le cadre d'un abonnement aux notifications, vous choisissez comment et où envoyer les notifications. Vous spécifiez le type de source dont vous voulez être informé. Actuellement, AWS DMS prend en charge les types de source d'instance de réplication et de tâche de réplication. De plus, selon le type de source que vous sélectionnez, vous choisissez les catégories d'évènements et vous identifiez la source pour laquelle vous souhaitez recevoir des notifications d'événement.

Utilisation de AWS Management Console

> **Important**
>
> Depuis la publication d'AWS DMS 3.4.5 et avec les versions ultérieures, nous vous recommandons d'utiliser Amazon EventBridge pour envoyer des notifications quand un événement AWS DMS se produit. Pour plus d'informations sur l'utilisation des événements EventBridge avec AWS DMS, consultez Utilisation des événements et des notifications Amazon EventBridge dans AWS Database Migration Service.

Pour vous abonner aux notifications d'événement AWS DMS avec Amazon SNS à l'aide de la console

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS DMS à l'adresse https://console.aws.amazon.com/dms/v2/.

   Si vous êtes connecté en tant qu'utilisateur IAM, vous devez détenir les autorisations appropriées pour accéder à AWS DMS.
2. Dans le panneau de navigation, choisissez Abonnements aux évènements.
3. Sur la page Abonnements aux événements, choisissez Créer un abonnement aux événements.
4. Sur la page Créer un abonnement aux événements, procédez comme suit :

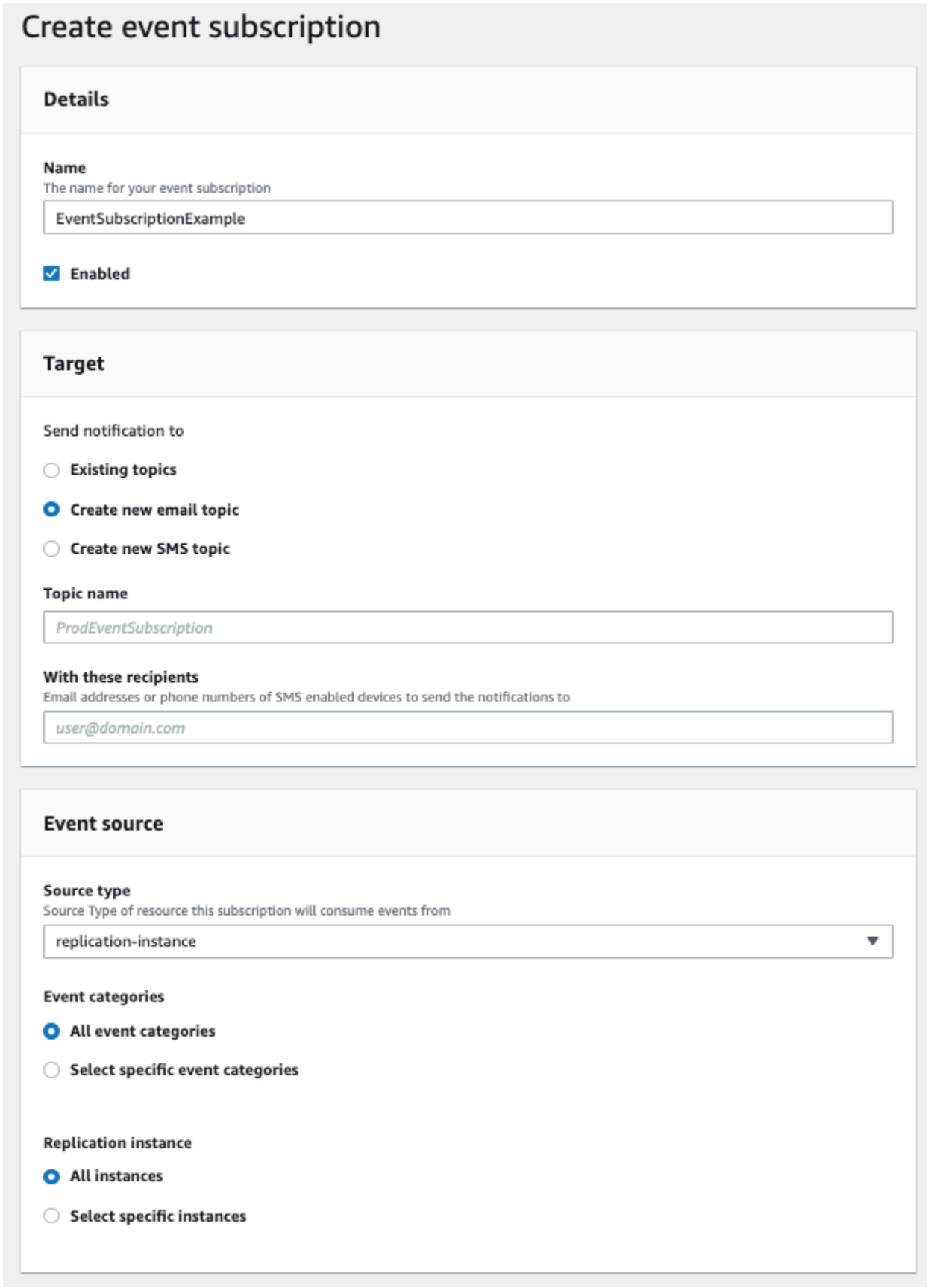
   a. Sous Détails, pour Nom, entrez un nom pour l'abonnement aux notifications d'événement.

   b. Choisissez Activé pour activer l'abonnement. Si vous souhaitez créer l'abonnement sans envoyer encore de notifications, ne choisissez pas Activé.

   c. Sous Cible, choisissez Rubriques existantes, Créer une rubrique d'e-mail ou Créer une rubrique SMS pour envoyer des notifications. Assurez-vous de disposer d'une rubrique Amazon SNS où envoyer les avis. Si besoin est, créez-en une. Si vous créez une rubrique, vous pouvez entrer une adresse e-mail à laquelle les notifications seront envoyées.

   d. Sous Origine de l'événement, pour Type de source, choisissez un type de source. Les seules options sont replication-instance et replication-task.

   e. Selon le type de source que vous avez sélectionné, choisissez les catégories d'événement et les sources pour lesquelles vous souhaitez recevoir des notifications d'événements.


Create event subscription
Details
Name
The name for your event subscription
EventSubscriptionExample
Enabled
Target
Send notification to
Existing topics
Create new email topic
Create new SMS topic
Topic name
ProdEventSubscription
With these recipients
Email addresses or phone numbers of SMS enabled devices to send the notifications to
user@domain.com
Event source
Source type
Source Type of resource this subscription will consume events from
replication-instance
Event categories
All event categories
Select specific event categories
Replication instance
All instances
Select specific instances

f. Sélectionnez Créer un abonnement aux événements.

La console AWS DMS indique que l'abonnement est en cours de création.

> **Note**
>
> Vous pouvez également créer des abonnements aux notifications d'événement Amazon SNS à l'aide de l'API et de la CLI AWS DMS. Pour plus d'informations, consultez CreateEventSubscription dans Référence d'API AWS DMS et create-event-subscription dans la documentation Référence de la CLI AWS DMS.

Validation de la stratégie d'accès de votre rubrique SNS

Votre stratégie d'accès SNS nécessite des autorisations permettant à AWS DMS de publier des événements dans votre rubrique SNS. Vous pouvez valider et mettre à jour votre stratégie d'accès comme décrit dans les procédures suivantes.

Pour valider votre stratégie d'accès

1. Ouvrez la console Amazon SNS.
2. Dans le volet de navigation, choisissez Rubriques et sélectionnez la rubrique pour laquelle vous souhaitez recevoir des notifications DMS.
3. Sélectionnez l'onglet Stratégie d'accès.

Vous pouvez mettre à jour votre stratégie si votre stratégie d'accès SNS n'autorise pas AWS DMS à publier des événements dans votre rubrique SNS.

Pour mettre à jour votre stratégie d'accès

1. Dans la section Détails de la page de votre rubrique, choisissez Modifier.
2. Développez la section Stratégie d'accès et attachez la stratégie suivante dans l'éditeur JSON.

```
{
      "Sid": "dms-allow-publish",
      "Effect": "Allow",
      "Principal": {
        "Service": "dms.amazonaws.com"
```

```
      },
      "Action": "sns:Publish",
      "Resource": "your-SNS-topic-ARN"
    }
```

Nous vous recommandons de restreindre davantage l'accès à votre rubrique SNS en spécifiant la condition `aws:SourceArn`, à savoir l'ARN de l'élément EventSubscription DMS qui publie les événements dans cette rubrique.

```
...
"Resource": "your-SNS-topic-ARN"
"Condition": {
    "StringEquals": {
       "aws:SourceArn": "arn:partition:dms:your-AWS-region:your-AWS-account-
ID:es:your-dms-es-arn or *"
 }
```

3. Choisissez Enregistrer les modifications.

Validation des données AWS DMS

Rubriques

- Statistiques des tâches de réplication
- Statistiques des tâches de réplication avec Amazon CloudWatch
- Revalidation de tables pendant une tâche
- Utilisation de l'éditeur JSON pour modifier les règles de validation
- Tâches de validation uniquement
- Résolution des problèmes
- Performances de validation Redshift
- Limites
- Validation des données cibles Amazon S3

AWS DMS assure la prise en charge de la validation des données, afin de garantir que vos données ont été migrées avec précision de la source vers la cible. Si elle est activée, la validation commence immédiatement après un chargement complet d'une table. La validation compare les modifications incrémentielles d'une tâche avec capture des données modifiées activée au fur et à mesure qu'elles apparaissent.

Lors de la validation des données, AWS DMS compare chaque ligne de la source avec sa ligne correspondante dans la cible, vérifie que les lignes contiennent les mêmes données et signale toute incohérence. Pour ce faire, AWS DMS émet les requêtes appropriées pour récupérer les données. Notez que ces requêtes consomment des ressources supplémentaires au niveau de la source et de la cible, ainsi que des ressources réseau supplémentaires.

Pour une tâche de CDC uniquement avec la validation activée, toutes les données préexistantes d'une table sont validées avant le début de la validation de nouvelles données.

La validation de données fonctionne avec les bases de données sources suivantes quand AWS DMS les prend en charge en tant que points de terminaison sources :

- Oracle
- Base de données compatible PostgreSQL (PostgreSQL, Aurora PostgreSQL ou Aurora sans serveur pour PostgreSQL)

- Base de données compatible MySQL (MySQL, MariaDB, Aurora MySQL ou Aurora sans serveur pour MySQL)
- Microsoft SQL Server
- IBM Db2 LUW

La validation des données fonctionne avec les bases de données cibles suivantes quand AWS DMS les prend en charge en tant que points de terminaison cibles :

- Oracle
- Base de données compatible PostgreSQL (PostgreSQL, Aurora PostgreSQL ou Aurora sans serveur pour PostgreSQL)
- Base de données compatible MySQL (MySQL, MariaDB, Aurora MySQL ou Aurora sans serveur pour MySQL)
- Microsoft SQL Server
- IBM Db2 LUW
- Amazon Redshift
- Amazon S3. Pour en savoir plus sur la validation des données cibles Amazon S3, consultez Validation des données cibles Amazon S3.

Pour plus d'informations sur les points de terminaison pris en charge, consultez Utilisation des points de terminaison AWS DMS.

La validation des données nécessite du temps supplémentaire, au-delà de celui requis pour la migration elle-même. Le temps supplémentaire requis dépend de la quantité de données ayant migré.

Pour plus d'informations sur ces paramètres, consultez la page Paramètres de la tâche de validation des données.

Pour obtenir un exemple de paramètres de tâche `ValidationSettings` dans un fichier JSON, consultez Exemple de paramètres de tâche.

Statistiques des tâches de réplication

Lorsque la validation des données est activée, AWS DMS fournit les statistiques suivantes au niveau de la table :

- ValidationState : état de validation de la table. Le paramètre peut avoir les valeurs suivantes :
  - Not enabled – La validation n'est pas activée pour la table de la tâche de migration.
  - Pending records – Certains enregistrements de la table sont en attente de validation.
  - Enregistrements non concordants : certains enregistrements de la table ne concordent pas entre la source et la cible. Une incohérence peut se produire pour diverses raisons. Pour plus d'informations, consultez la table `awsdms_control.awsdms_validation_failures_v1` sur le point de terminaison cible.
  - Suspended records (Enregistrements suspendus) – Certains enregistrements de la table ne peuvent pas être validés.
  - No primary key (Aucune clé primaire) – La table ne peut pas être validée, car elle n'avait pas de clé primaire.
  - Table error (Erreur de table) – La table n'a pas été validée, car elle était dans un état d'erreur et certaines données n'ont pas été migrées.
  - Validé : toutes les lignes de la table sont validées. Si la table est mise à jour, l'état peut devenir Validated.
  - Error (Erreur) – La table ne peut pas être validée en raison d'une erreur inattendue.
  - En attente de validation : la table attend sa validation.
  - Préparation de la table : préparation de la table activée dans la tâche de migration pour validation.
  - En attente de revalidation : toutes les lignes de la table sont en attente de validation après la mise à jour de la table.
- ValidationPending (Validation en attente) – Nombre d'enregistrements qui ont été migrés vers la cible, mais qui n'ont pas encore été validés.
- ValidationSuspended – Nombre d'enregistrements qu'AWS DMS ne peut pas comparer. Par exemple, si un enregistrement de la source est constamment mis à jour, AWS DMS ne peut pas comparer la source et la cible.
- ValidationFailed : nombre d'enregistrements qui n'ont pas réussi la phase de validation des données.

Pour obtenir un exemple de paramètres de tâche `ValidationSettings` dans un fichier JSON, consultez Exemple de paramètres de tâche.

Vous pouvez consulter les informations de validation des données à l'aide de la console, d'AWS CLI ou de l'API AWS DMS.

- Sur la console, vous pouvez choisir de valider une tâche lorsque vous créez ou modifiez la tâche. Pour afficher le rapport de validation des données à l'aide de la console, choisissez la tâche sur la page Tâches et choisissez l'onglet Statistiques de table dans la section des détails.
- À l'aide de l'interface de ligne de commande, définissez le paramètre `EnableValidation` sur `true` lorsque vous créez ou modifiez une tâche pour commencer la validation des données. L'exemple suivant crée une tâche et active la validation des données.

```
create-replication-task
  --replication-task-settings '{"ValidationSettings":{"EnableValidation":true}}'
  --replication-instance-arn arn:aws:dms:us-east-1:5731014:
     rep:36KWVMB7Q
  --source-endpoint-arn arn:aws:dms:us-east-1:5731014:
     endpoint:CSZAEFQURFYMM
  --target-endpoint-arn arn:aws:dms:us-east-1:5731014:
     endpoint:CGPP7MF6WT4JQ
  --migration-type full-load-and-cdc
  --table-mappings '{"rules": [{"rule-type": "selection", "rule-id": "1",
     "rule-name": "1", "object-locator": {"schema-name": "data_types", "table-name":
 "%"},
     "rule-action": "include"}]}'
```

Utilisez la commande `describe-table-statistics` pour recevoir le rapport de validation des données au format JSON. La commande suivante affiche le rapport de validation des données.

```
aws dms  describe-table-statistics --replication-task-arn arn:aws:dms:us-
east-1:5731014:
rep:36KWVMB7Q
```

Le rapport doit se présenter comme suit :

```
{
    "ReplicationTaskArn": "arn:aws:dms:us-west-2:5731014:task:VFPFTYKK2RYSI",
    "TableStatistics": [
        {
            "ValidationPendingRecords": 2,
            "Inserts": 25,
            "ValidationState": "Pending records",
            "ValidationSuspendedRecords": 0,
            "LastUpdateTime": 1510181065.349,
            "FullLoadErrorRows": 0,
```

```
            "FullLoadCondtnlChkFailedRows": 0,
            "Ddls": 0,
            "TableName": "t_binary",
            "ValidationFailedRecords": 0,
            "Updates": 0,
            "FullLoadRows": 10,
            "TableState": "Table completed",
            "SchemaName": "d_types_s_sqlserver",
            "Deletes": 0
        }
}
```

- À l'aide de l'API AWS DMS, créez une tâche utilisant l'action CreateReplicationTask et définissez le paramètre `EnableValidation` sur true pour valider les données qui ont été migrées par la tâche. Utilisez l'action DescribeTableStatistics pour recevoir le rapport de validation des données au format JSON.

Statistiques des tâches de réplication avec Amazon CloudWatch

Quand Amazon CloudWatch est activé, AWS DMS fournit les statistiques suivantes sur les tâches de réplication :

- ValidationSucceededRecordCount : nombre de lignes validées par minute par AWS DMS.
- ValidationAttemptedRecordCount - Nombre de lignes pour lesquelles la validation a été tentée, par minute.
- ValidationFailedOverallCount - Nombre de lignes pour lesquelles la validation a échoué.
- ValidationSuspendedOverallCount - Nombre de lignes pour lesquelles la validation a été suspendue.
- ValidationPendingOverallCount - Nombre de lignes pour lesquelles la validation est toujours en attente.
- ValidationBulkQuerySourceLatency : AWS DMS peut effectuer la validation en bloc des données, en particulier dans certains scénarios lors d'un chargement complet ou d'une réplication continue avec de nombreuses modifications. Cette métrique indique le temps de latence requis pour lire un ensemble de données à partir du point de terminaison source.
- ValidationBulkQueryTargetLatency : AWS DMS peut effectuer la validation en bloc des données, en particulier dans certains scénarios lors d'un chargement complet ou d'une réplication continue

avec de nombreuses modifications. Cette métrique indique le temps de latence requis pour lire un ensemble de données en direction du point de terminaison cible.

- ValidationItemQuerySourceLatency : au cours de la réplication continue, la validation des données peut identifier les modifications en cours et les valider. Cette métrique indique le temps de latence lors de la lecture de ces modifications à partir de la source. La validation peut exécuter plus de requêtes que nécessaire, en fonction du nombre de modifications, si des erreurs se produisent lors de la validation.
- ValidationItemQueryTargetLatency : au cours de la réplication continue, la validation des données peut identifier les modifications en cours et les valider ligne par ligne. Cette métrique indique le temps de latence lors de la lecture de ces modifications à partir de la cible. La validation peut exécuter plus de requêtes que nécessaire, en fonction du nombre de modifications, si des erreurs se produisent lors de la validation.

Pour collecter des informations de validation des données à partir des statistiques activées par CloudWatch, sélectionnez Activer CloudWatch Logs lorsque vous créez ou modifiez une tâche à l'aide de la console. Ensuite, pour consulter les informations de validation des données et garantir que vos données ont été migrées avec précision depuis la source vers la cible, procédez comme suit.

1. Choisissez la tâche sur la page Tâches de migration de base de données.
2. Choisissez l'onglet Métriques CloudWatch.
3. Sélectionnez Validation dans le menu déroulant.

Revalidation de tables pendant une tâche

Pendant l'exécution d'une tâche, vous pouvez demander à AWS DMS d'effectuer une validation des données.

AWS Management Console

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS DMS à l'adresse https://console.aws.amazon.com/dms/v2/.

   Si vous êtes connecté en tant qu'utilisateur AWS Identity and Access Management (IAM), veillez à disposer des autorisations appropriées pour accéder à AWS DMS. Pour connaître les autorisations requises, consultez Autorisations IAM nécessaires pour utiliser AWS DMS.
2. Choisissez Tâches dans le volet de navigation.

3. Choisissez la tâche en cours d'exécution contenant la table à revalider.
4. Choisissez l'onglet Statistiques de table.
5. Choisissez la table à revalider (vous pouvez choisir jusqu'à 10 tables en même temps). Si la tâche n'est plus en cours d'exécution, vous ne pouvez pas revalider la ou les tables.
6. Choisissez Revalider.

Utilisation de l'éditeur JSON pour modifier les règles de validation

Pour ajouter une règle de validation à une tâche à l'aide de l'éditeur JSON depuis la console AWS DMS, procédez comme suit :

1. Sélectionnez Tâches de migration de base de données.
2. Sélectionnez votre tâche dans la liste des tâches de migration.
3. Si votre tâche est en cours d'exécution, sélectionnez Arrêter dans le menu déroulant Actions.
4. Une fois la tâche arrêtée, pour modifier votre tâche, sélectionnez Modifier dans le menu déroulant Actions.
5. Dans la section Mappages de table, sélectionnez Éditeur JSON et ajoutez votre règle de validation à vos mappages de tables.

Par exemple, vous pouvez ajouter la règle de validation suivante pour exécuter une fonction de remplacement sur la source. Dans ce cas, si la règle de validation rencontre un octet null, elle le valide en tant qu'espace.

```
{
 "rule-type": "validation",
 "rule-id": "1",
 "rule-name": "1",
 "rule-target": "column",
 "object-locator": {
  "schema-name": "Test-Schema",
  "table-name": "Test-Table",
  "column-name": "Test-Column"
 },
 "rule-action": "override-validation-function",
 "source-function": "REPLACE(${column-name}, chr(0), chr(32))",
 "target-function": "${column-name}"
}
```

Tâches de validation uniquement

Vous pouvez créer des tâches de validation uniquement pour prévisualiser et valider les données sans effectuer de migration ni de réplication de données. Pour créer une tâche de validation uniquement, définissez les paramètres `EnableValidation` et `ValidationOnly` sur `true`. Lors de l'activation de `ValidationOnly`, des exigences supplémentaires s'appliquent. Pour de plus amples informations, veuillez consulter Paramètres de la tâche de validation des données.

Pour un type de migration à chargement complet uniquement, une tâche de validation uniquement s'exécute beaucoup plus rapidement que son équivalent CDC lorsque de nombreux échecs sont signalés. Toutefois, les modifications apportées au point de terminaison source ou cible sont signalées comme des échecs en mode de chargement complet, ce qui peut constituer un inconvénient.

Une tâche de validation CDC uniquement retarde la validation en fonction de la latence moyenne et effectue de nouvelles tentatives en cas d'échecs avant de les signaler. Si la majorité des comparaisons de données aboutissent à des échecs, une tâche de validation uniquement pour le mode CDC est très lente, ce qui constitue un inconvénient potentiel.

Une tâche de validation uniquement doit être configurée dans le même sens que la tâche de réplication, en particulier pour la CDC. Cela est dû au fait qu'une tâche de validation CDC uniquement détecte les lignes qui ont changé et doivent être revalidées en fonction du journal des modifications de la source. Si la cible est spécifiée en tant que source, elle ne connaît que les modifications envoyées à la cible par DMS et il n'est pas garanti qu'elle détecte les erreurs de réplication.

Validation du chargement complet uniquement

À partir de AWS DMS version 3.4.6, une tâche de validation de chargement complet uniquement compare rapidement toutes les lignes des tables source et cible en un seul passage, signale immédiatement tous les échecs, puis s'arrête. La validation n'est jamais suspendue en raison d'échecs dans ce mode, elle est optimisée pour la vitesse. Toutefois, les modifications apportées au point de terminaison source ou cible sont signalées comme des échecs.

> **Note**
>
> À partir de AWS DMS version 3.4.6, ce comportement de validation s'applique également à la tâche de migration à chargement complet avec validation activée.

Validation CDC uniquement

Une tâche de validation CDC uniquement valide toutes les lignes existantes entre les tables source et cible lors d'un nouveau départ. En outre, une tâche de validation CDC uniquement s'exécute en continu, revalide les modifications de réplication continue, limite le nombre d'échecs signalés à chaque passage et effectue de nouvelles tentatives avec les lignes non concordantes avant de signaler un échec. Elle est optimisée pour éviter les faux positifs.

La validation d'une table (ou de l'ensemble de la tâche) est suspendue si les seuils `FailureMaxCount` ou `TableFailureMaxCount` sont dépassés. Cela s'applique également à une tâche de migration CDC ou Chargement complet+CDC avec la validation activée. De plus, une tâche CDC avec la validation activée retarde la revalidation pour chaque ligne modifiée en fonction de la latence moyenne sur la source et la cible.

Toutefois, une tâche de validation CDC uniquement ne migre pas de données et n'a pas de latence. Elle définit `ValidationQueryCdcDelaySeconds` sur 180 par défaut. Vous pouvez également augmenter ce montant pour tenir compte des environnements à latence élevée et éviter les faux positifs.

Cas d'utilisation relatifs à la validation uniquement

Les cas d'utilisation pour scinder la partie validation des données d'une tâche de migration ou de réplication en une tâche de validation uniquement distincte incluent, sans toutefois s'y limiter, les cas suivants :

- Contrôlez exactement le moment où la validation a lieu : les requêtes de validation ajoutent une charge supplémentaire aux points de terminaison sources et cibles. Il peut donc être avantageux de migrer ou de répliquer les données d'abord dans une tâche, puis de valider les résultats dans une autre tâche.
- Réduisez la charge sur l'instance de réplication : il peut être avantageux de diviser la validation des données pour l'exécuter sur sa propre instance.

- Obtenez rapidement le nombre de lignes qui ne correspondent pas à un moment donné : par exemple, juste avant ou pendant une interruption de production pendant une fenêtre de maintenance par rapport à un point de terminaison cible, vous pouvez créer une tâche de validation de chargement complet uniquement pour obtenir une réponse à votre question.
- Lorsque des échecs de validation sont attendus pour une tâche de migration avec un composant CDC : par exemple, lors de la migration de données `varchar2` Oracle vers `jsonb` PostgreSQL, la validation CDC réessaie sans cesse ces lignes qui ont échoué et limite le nombre d'échecs signalés chaque fois. Mais vous pouvez créer une tâche de validation de chargement complet uniquement et obtenir une réponse plus rapide.
- Vous avez développé un script/utilitaire de réparation de données qui lit la table des échecs de validation : (voir également, Résolution des problèmes). Une tâche de validation de chargement complet uniquement signale rapidement les échecs afin que le script de réparation des données puisse agir.

Pour obtenir un exemple de paramètres de tâche `ValidationSettings` dans un fichier JSON, consultez Exemple de paramètres de tâche.

Résolution des problèmes

Pendant la validation, AWS DMS crée une nouvelle table au point de terminaison de la cible : `awsdms_control.awsdms_validation_failures_v1`. Si l'état d'un enregistrement devient ValidationSuspended ou ValidationFailed, AWS DMS écrit les informations de diagnostic sur `awsdms_control.awsdms_validation_failures_v1`. Vous pouvez interroger cette table pour aider à résoudre les erreurs de validation.

Pour en savoir plus sur la modification du schéma par défaut dans lequel la table est créée sur la cible, consultez Paramètres de tâche de la table de contrôle.

Voici une description de la table `awsdms_control.awsdms_validation_failures_v1` :

Nom de la colonne	Type de données	Description
`TASK_NAME`	`VARCHAR(128) NOT NULL`	Identificateur de tâche AWS DMS.

Nom de la colonne	Type de données	Description
TABLE_OWNER	VARCHAR(128) NOT NULL	Schéma (propriétaire) de la table.
TABLE_NAME	VARCHAR(128) NOT NULL	Nom de la table.
FAILURE_TIME	DATETIME(3) NOT NULL	Heure où la défaillance s'est produite.
KEY_TYPE	VARCHAR(128) NOT NULL	Réservé pour une utilisation future (la valeur est toujours « Row »)
KEY	TEXT NOT NULL	Il s'agit de la clé primaire du type Row Record.
FAILURE_TYPE	VARCHAR(128) NOT NULL	Gravité de l'erreur de validation. Peut avoir la valeur RECORD_DIFF , MISSING_SOURCE ou MISSING_TARGET .
DETAILS	VARCHAR(8000) NOT NULL	Chaîne au format JSON contenant toutes les valeurs des colonnes source/cible qui ne correspondent pas pour la clé donnée.

La requête suivante affiche toutes les défaillances d'une tâche en interrogeant la table awsdms_control.awsdms_validation_failures_v1. Le nom de la tâche doit être l'ID de ressource externe de la tâche. L'ID de ressource externe de la tâche est la dernière valeur de l'ARN de la tâche. Par exemple, pour une tâche avec la valeur d'ARN arn:aws:dms:us-west-2:5599:task: VFPFKH4FJR3FTYKK2RYSI, l'ID de ressource externe de la tâche sera VFPFKH4FJR3FTYKK2RYSI.

```
select * from awsdms_validation_failures_v1 where TASK_NAME = 'VFPFKH4FJR3FTYKK2RYSI'

TASK_NAME       VFPFKH4FJR3FTYKK2RYSI
TABLE_OWNER     DB2PERF
TABLE_NAME      PERFTEST
FAILURE_TIME    2020-06-11 21:58:44
```

```
KEY_TYPE        Row
KEY             {"key":  ["3451491"]}
FAILURE_TYPE    RECORD_DIFF
DETAILS         [[{'MYREAL': '+1.10106036e-01'}, {'MYREAL': '+1.10106044e-01'}],]
```

Vous pouvez examiner le champ `DETAILS` pour déterminer quelles colonnes ne correspondent pas. Comme vous avez la clé primaire de l'enregistrement ayant échoué, vous pouvez interroger les points de terminaison sources et cibles pour voir quelle partie de l'enregistrement ne correspond pas.

Performances de validation Redshift

Amazon Redshift se distingue des bases de données relationnelles à plusieurs égards : stockage en colonnes, MPP, compression des données, etc. Ces différences confèrent à Redshift un profil de performance différent de celui des bases de données relationnelles.

Pendant la phase de réplication à chargement complet, la validation utilise des requêtes d'intervalle, la taille des données étant régie par le paramètre `PartitionSize`. Ces requêtes d'intervalle sélectionnent tous les enregistrements de la table source.

Pour une réplication continue, les requêtes passent d'une recherche par intervalle à une recherche par enregistrement. Le type de requête est déterminé dynamiquement en fonction de plusieurs facteurs, tels que :

- Volume des requêtes
- Types de requêtes DML sur la table source
- Latence des tâches
- Nombre total d'enregistrements
- Paramètres de validation tels que `PartitionSize`

Il est possible que votre cluster Amazon Redshift soit soumis à une charge supplémentaire en raison des requêtes de validation. Comme les facteurs ci-dessus varient selon les cas d'utilisation, vous devez passer en revue les performances de vos requêtes de validation et ajuster votre cluster et votre table en conséquence. Parmi les options permettant d'atténuer les problèmes de performances, citons les suivantes :

- Réduisez les paramètres `PartitionSize` et `ThreadCount` pour réduire la charge de travail pendant la validation à chargement complet. Notez que la validation des données sera ralentie.
- Bien que Redshift n'applique pas de clés primaires, AWS DMS s'appuie sur des clés primaires pour identifier de manière unique les enregistrements sur la cible à des fins de validation des données. Si possible, définissez la clé primaire de manière à ce qu'elle corresponde à la clé de tri, afin que les requêtes de validation à chargement complet s'exécutent plus rapidement.

Limites

- La validation des données exige que la table dispose d'une clé primaire ou d'un index unique.
  - Les colonnes de clé primaire ne peuvent pas être de type `CLOB`, `BLOB` ou `BYTE`.
  - Pour les colonnes de clé primaire de type `VARCHAR` ou `CHAR`, la longueur doit être inférieure à 1 024. Vous devez spécifier la longueur du type de données. Vous ne pouvez pas utiliser de types de données de longueur illimitée comme clé primaire pour la validation des données.
  - Une clé Oracle créée avec la clause `NOVALIDATE` n'est pas considérée comme une clé primaire ou un index unique.
  - Pour une table Oracle sans clé primaire et uniquement avec une clé unique, les colonnes avec la contrainte unique doivent également avoir une contrainte `NOT NULL`.
- La validation des valeurs NULL PK/UK n'est pas prise en charge.
- Si le classement de la colonne de clé primaire dans l'instance PostgreSQL cible n'est pas défini sur « C », l'ordre de tri de la clé primaire est différent de celui d'Oracle. Si l'ordre de tri est différent dans PostgreSQL et dans Oracle, la validation des données ne parvient pas à valider les enregistrements.
- La validation des données génère des requêtes supplémentaires sur les bases de données source et cible. Vous devez vous assurer que les deux bases de données aient des ressources suffisantes pour gérer cette charge supplémentaire. Cela est particulièrement vrai pour les cibles Redshift. Pour plus d'informations, consultez Performances de validation Redshift ci-après.
- La validation des données n'est pas prise en charge si plusieurs bases de données sont consolidées en une seule.
- Pour un point de terminaison Oracle source ou cible, AWS DMS utilise DBMS_CRYPTO pour valider les LOB. Si votre point de terminaison Oracle utilise les LOB, vous devez accorder l'autorisation d'exécution sur dbms_crypto au compte d'utilisateur qui permet d'accéder au point de terminaison Oracle. Vous pouvez le faire en exécutant l'instruction suivante :

```
grant execute on sys.dbms_crypto to dms_endpoint_user;
```

- Si la base de données cible est modifiée en dehors d'AWS DMS pendant la validation, les écarts peuvent ne pas être signalés avec précision. Ce résultat peut se produire si l'une de vos applications écrit des données dans la table cible, tandis qu'AWS DMS exécute la validation sur cette même table.
- Si une ou plusieurs lignes sont modifiées de façon continue au cours de la validation, AWS DMS ne peut pas valider ces lignes.
- Si AWS DMS détecte plus de 10 000 enregistrements suspendus ou ayant échoué, il arrête la validation. Avant de poursuivre, vous devez résoudre les problèmes sous-jacents relatifs aux données.
- AWS DMS ne prend pas en charge la validation des données des vues.
- AWS DMS ne prend pas en charge la validation des données lorsque les paramètres de tâche de substitution de caractères sont utilisés.
- AWS DMS ne prend pas en charge la validation du type Oracle LONG.
- AWS DMS ne prend pas en charge la validation du type Oracle Spatial lors d'une migration hétérogène.

Pour connaître les limites d'utilisation de la validation des cibles S3, consultez Limitations liées à l'utilisation de la validation des cibles S3.

Validation des données cibles Amazon S3

AWS DMS prend en charge la validation des données répliquées dans les cibles Amazon S3. Comme AWS DMS stocke les données répliquées sous forme de fichiers plats dans Amazon S3, nous utilisons des requêtes Amazon Athena `CREATE TABLE AS SELECT` (CTAS) pour valider les données.

Les requêtes sur les données stockées dans Amazon S3 nécessitent beaucoup de ressources informatiques. Ainsi, AWS DMS exécute la validation des données Amazon S3 lors de la capture des données de modification (CDC) une seule fois par jour, à minuit (00:00) UTC. Chaque validation quotidienne exécutée par AWS DMS est appelée validation par intervalle. Au cours d'une validation par intervalle, AWS DMS valide tous les enregistrements de modifications qui ont été migrés vers le compartiment Amazon S3 cible au cours des dernières 24 heures. Pour plus d'informations sur les

limitations de validation par intervalle, consultez Limitations liées à l'utilisation de la validation des cibles S3.

La validation des cibles Amazon S3 utilise Amazon Athena, ce qui occasionne des frais supplémentaires. Pour plus d'informations, consultez Tarification Amazon Athena.

> **Note**
>
> La validation des cibles S3 nécessite AWS DMS version 3.5.0 ou ultérieure.

Rubriques

- Conditions préalables à la validation des cibles S3
- Autorisations d'utilisation de la validation des cibles S3
- Limitations liées à l'utilisation de la validation des cibles S3
- Utilisation de tâches de validation uniquement avec la validation des cibles S3

Conditions préalables à la validation des cibles S3

Avant d'utiliser la validation des cibles S3, vérifiez les paramètres et autorisations suivants :

- Définissez la valeur de `DataFormat` dans les paramètres S3Settings du point de terminaison sur `parquet`. Pour plus d'informations, consultez Paramètres Parquet pour S3.
- Veillez à ce que le rôle affecté au compte d'utilisateur utilisé pour créer la tâche de migration dispose de l'ensemble d'autorisations correct. Consultez Autorisations ci-dessous.

Pour les tâches utilisant la réplication continue (CDC), vérifiez les paramètres suivants :

- Activez la journalisation supplémentaire afin de disposer d'enregistrements complets dans les données de CDC. Pour en savoir plus sur l'activation de la journalisation supplémentaire, consultez Ajout automatique d'une journalisation supplémentaire à un point de terminaison source Oracle dans la section Résolution des problèmes et assistance au diagnostic de ce guide.
- Définissez le paramètre `TimestampColumnName` du point de terminaison cible. Le nom de la colonne d'horodatage n'est soumis à aucune limitation. Pour plus d'informations, consultez S3Settings.

- Configurez le partitionnement des dossiers basé sur la date pour la cible. Pour plus d'informations, consultez Utilisation du partitionnement de dossiers basé sur la date.

Autorisations d'utilisation de la validation des cibles S3

Pour configurer l'accès afin d'utiliser la validation des cibles S3, veillez à ce que le rôle affecté au compte d'utilisateur utilisé pour créer la tâche de migration dispose de l'ensemble d'autorisations suivant. Remplacez les exemples de valeurs par vos valeurs.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "athena:StartQueryExecution",
                "athena:GetQueryExecution",
                "athena:CreateWorkGroup"
            ],
            "Resource": "arn:aws:athena:<endpoint_region_code>:<account_id>:workgroup/
dms_validation_workgroup_for_task_*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateDatabase",
                "glue:DeleteDatabase",
                "glue:GetDatabase",
                "glue:GetTables",
                "glue:CreateTable",
                "glue:DeleteTable",
                "glue:GetTable"
            ],
            "Resource": [
                "arn:aws:glue:<endpoint_region_code>:<account_id>:catalog",
                "arn:aws:glue:<endpoint_region_code>:<account_id>:database/
aws_dms_s3_validation_*",
                "arn:aws:glue:<endpoint_region_code>:<account_id>:table/
aws_dms_s3_validation_*/*",
                "arn:aws:glue:<endpoint_region_code>:<account_id>:userDefinedFunction/
aws_dms_s3_validation_*/*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucketMultipartUploads",
                "s3:AbortMultipartUpload",
                "s3:ListMultipartUploadParts"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket_name>",
                "arn:aws:s3:::<bucket_name>/*"
            ]
        }
    ]
}
```

Limitations liées à l'utilisation de la validation des cibles S3

Consultez les limitations supplémentaires suivantes qui s'appliquent lors de l'utilisation de la validation des cibles S3. Pour connaître les limitations qui s'appliquent à toutes les validations, consultez Limites.

- Votre valeur `DatePartitionSequence` a besoin d'une composante de jour. La validation des cibles S3 ne prend pas en charge le format `YYYYMM`.
- Quand la validation par intervalle est en cours pendant la CDC, vous pouvez voir de fausses erreurs de validation dans la table `awsdms_validation_failures_v1`. Ces erreurs se produisent car AWS DMS migre les modifications qui sont arrivées pendant l'intervalle de validation vers le dossier de partitionnement du jour suivant. Normalement, ces modifications sont écrites dans le dossier de partitionnement du jour actuel. Ces fausses erreurs limitent la validation de la réplication depuis une base de données source dynamique vers une cible statique, telle qu'Amazon S3. Pour mener des investigations sur ces fausses erreurs, examinez les enregistrements situés près de la fin de la fenêtre de validation (00:00 UTC), où ces erreurs apparaissent généralement.

Pour minimiser le nombre de fausses erreurs, veillez à ce que la valeur `CDCLatencySource` de la tâche soit faible. Pour en savoir plus sur la surveillance de la latence, consultez Métriques de tâches de réplication.

- Les tâches à l'état `failed` ou `stopped` ne valident pas les modifications de la veille. Pour minimiser les erreurs de validation dues à des échecs inattendus, créez des tâches distinctes de validation uniquement avec les mêmes mappages de tables et les points de terminaison sources et cibles. Pour plus d'informations sur les tâches de validation uniquement, consultez Utilisation de tâches de validation uniquement avec la validation des cibles S3.
- La colonne Statut de la validation, dans les statistiques de table, reflète l'état de la dernière validation par intervalle. Par conséquent, une table présentant des incohérences peut apparaître comme validée après la validation par intervalle du jour suivant. Vérifiez le `s3_validation_failures folder` dans le compartiment Amazon S3 cible et recherchez des incohérences survenues il y a plus d'un jour.
- S3 Validation utilise la fonctionnalité de table à compartiments d'Amazon Athena. Cela permet à la validation S3 de créer une copie groupée des données de la table cible. Cela signifie que la copie des données de la table est divisée en sous-ensembles correspondant au partitionnement interne de la validation DMS. Les tables à godets Athena ont une limite de 100 000 seaux. Toute table que la validation S3 tente de valider et qui dépasse cette limite échouera à la validation. Le nombre de compartiments que S3 Validation tente de créer est égal au nombre suivant :

  ```
  (#records in the table) / (validation partition size setting)
  ```

  Pour contourner cette limitation, augmentez le paramètre de taille de partition de validation afin que le nombre de compartiments créés par S3 Validation soit inférieur à 100 000. Pour plus d'informations sur le partitionnement, consultez la section Partitionnement et découpage dans Athena dans le guide de l'utilisateur d'Amazon Athena.

Utilisation de tâches de validation uniquement avec la validation des cibles S3

Une tâche de validation uniquement exécute la validation sur les données à migrer sans exécuter la migration.

Les tâches de validation uniquement continuent de s'exécuter, même si la tâche de migration s'arrête, ce qui garantit qu'AWS DMS ne manquera pas la fenêtre de validation par intervalle à 00 h 00 UTC.

L'utilisation de tâches de validation uniquement avec les points de terminaison cibles Amazon S3 présente les limitations suivantes :

- La validation Amazon S3 pour les tâches de chargement complet avec le paramètre de validation uniquement activé est prise en charge, mais fonctionne différemment des tâches de chargement complet et de validation uniquement pour les autres points de terminaison. Pour S3 en tant que cible, une tâche de ce type valide uniquement par rapport aux données de chargement complet dans la cible S3 et ne valide pas par rapport à des données migrées dans le cadre d'une migration CDC. Utilisez cette fonctionnalité uniquement pour valider les données créées par une tâche de chargement complet uniquement. L'utilisation de ce mode pour valider les données d'une cible sur laquelle une tâche CDC active s'exécute ne produira pas une validation efficace.
- Les tâches de validation uniquement valident uniquement les modifications effectuées depuis la dernière fenêtre de validation par intervalle (00:00 UTC). Les tâches de validation uniquement ne valident pas les données de chargement complet ni les données de CDC des jours précédents.

Balisage des ressources dans AWS Database Migration Service

Vous pouvez utiliser des balises dans AWS Database Migration Service (AWS DMS) pour ajouter des métadonnées à vos ressources. De plus, ces balises peuvent être utilisées avec des politiques AWS Identity and Access Management (IAM) pour gérer l'accès aux ressources AWS DMS et contrôler les actions qui peuvent être appliquées aux ressources AWS DMS. Enfin, vous pouvez utiliser ces balises pour suivre les coûts en regroupant les dépenses pour des ressources balisées de la même façon.

Toutes les ressources AWS DMS peuvent être balisées :

- Certificats
- Fournisseurs de données
- Migrations de données
- Points de terminaison
- Abonnements aux événements
- Profils d'instance
- Projets de migration
- Instances de réplication
- Groupes de sous-réseaux de réplication
- Tâches de réplication

Une balise AWS DMS est une paire nom-valeur que vous définissez et associez à une ressource AWS DMS. Le nom s'appelle la clé. Fournir une valeur pour la clé est facultatif. Vous pouvez utiliser des balises pour affecter des informations arbitraires à une ressource AWS DMS. Une clé de balise peut être utilisée, par exemple, pour définir une catégorie, et la valeur de balise peut être un élément de cette catégorie. Par exemple, vous pouvez définir une clé de balise « projet » et une valeur de balise « Salix », indiquant que la ressource AWS DMS est affectée au projet Salix. Vous pouvez également utiliser des balises pour indiquer que des ressources AWS DMS sont utilisées pour des tests ou en production en utilisant une clé telle que environnement=test ou environnement =production. Nous vous recommandons d'utiliser un ensemble cohérent de clés de balise pour faciliter le suivi des métadonnées associées aux ressources AWS DMS.

Utilisez des étiquettes pour organiser votre facture AWS afin de refléter votre propre structure de coût. Pour ce faire, inscrivez-vous pour obtenir votre facture Compte AWS avec les valeurs de clé de balise incluses. Ensuite, pour voir le coût de vos ressources combinées, organisez vos informations de facturation en fonction des ressources possédant les mêmes valeurs de clé de balise. Par exemple, vous pouvez baliser plusieurs ressources avec un nom d'application spécifique, puis organiser vos informations de facturation pour afficher le coût total de cette application dans plusieurs services. Pour de plus amples informations, veuillez consulter Utilisation des balises d'allocation des coûts dans le Guide de l'utilisateur AWS Billing.

Chaque ressource AWS DMS possède un ensemble de balises qui contient toutes les balises affectées à cette ressource AWS DMS. Un ensemble de balises peut contenir jusqu'à dix balises ou n'en contenir aucune. Si vous ajoutez une balise à une ressource AWS DMS ayant la même clé qu'une balise existante sur cette ressource, la nouvelle valeur remplace l'ancienne.

AWS n'applique aucune signification sémantique aux balises ; celles-ci sont interprétées strictement comme des chaînes de caractères. AWS DMS peut définir des balises sur une ressource AWS DMS, en fonction des paramètres que vous utilisez lorsque vous créez la ressource.

La liste ci-dessous décrit les caractéristiques d'une balise AWS DMS.

- La clé de balise correspond au nom obligatoire de la balise. La valeur de la chaîne peut comporter de 1 à 128 caractères Unicode et ne peut pas être précédée de « aws: » ou « dms: ». La chaîne peut uniquement contenir l'ensemble de lettres, de chiffres et d'espaces Unicode, « _ », « . », « / », « = », « + », « - » (regex Java : `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`).

- La valeur de balise correspond à la valeur de chaîne facultative de la balise. La valeur de la chaîne peut comporter de 1 à 256 caractères Unicode et ne peut pas être précédée de « aws: » ou « dms: ». La chaîne peut uniquement contenir l'ensemble de lettres, de chiffres et d'espaces Unicode, « _ », « . », « / », « = », « + », « - » (regex Java : `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`).

  Les valeurs comprises dans un ensemble de balises ne doivent pas nécessairement être uniques et peuvent être null. Par exemple, vous pouvez avoir une paire clé-valeur dans un ensemble de balises appelé projet/Trinity et centre-de-coûts/Trinity.

Vous pouvez utiliser AWS CLI ou l'API AWS DMS pour ajouter, répertorier et supprimer les balises sur les ressources AWS DMS. Lorsque vous utilisez AWS CLI ou l'API AWS DMS, vous devez fournir l'Amazon Resource Name (ARN) pour la ressource AWS DMS avec laquelle vous souhaitez

travailler. Pour plus d'informations sur la création d'un ARN, consultez Création d'un nom de ressource Amazon (ARN) pour AWS DMS.

Notez que les balises sont mises en cache à des fins d'autorisation. Pour cette raison, les ajouts et les mises à jour des balises sur les ressources AWS DMS peuvent prendre plusieurs minutes avant d'être disponibles.

API

Vous pouvez ajouter, répertorier ou supprimer des balises pour une ressource AWS DMS à l'aide de l'API AWS DMS.

- Pour ajouter une balise à une ressource AWS DMS, utilisez l'opération `AddTagsToResource`.
- Pour répertorier les balises affectées à une ressource AWS DMS, utilisez l'opération `ListTagsForResource`.
- Pour supprimer des balises d'une ressource AWS DMS, utilisez l'opération `RemoveTagsFromResource`.

Pour en savoir sur la création de l'ARN requis, consultez Création d'un nom de ressource Amazon (ARN) pour AWS DMS.

Lorsque vous travaillez avec XML à l'aide de l'API AWS DMS, les balises utilisent le schéma suivant :

```
<Tagging>
  <TagSet>
   <Tag>
    <Key>Project</Key>
    <Value>Trinity</Value>
   </Tag>
   <Tag>
    <Key>User</Key>
    <Value>Jones</Value>
   </Tag>
  </TagSet>
</Tagging>
```

Le tableau suivant fournit une liste des balises XML autorisées et leurs caractéristiques. Notez que les valeurs pour la clé et la valeur sont sensibles à la casse. Par exemple, projet=Trinity et PROJET=Trinity sont deux balises différentes.

Elément de balisage	Description
TagSet	Un ensemble de balises contient toutes les balises assignées à une ressource Amazon RDS. Il ne peut y avoir qu'un ensemble de balises par ressource. Vous travaillez avec un TagSet uniquement via l'API AWS DMS.
Tag	Une balise est une paire clé-valeur définie par l'utilisateur. Il peut y avoir de 1 à 10 balises dans un ensemble de balises.
Clé	Une clé est le nom obligatoire de la balise. La valeur de la chaîne peut comporter de 1 à 128 caractères Unicode et ne peut pas être précédée de « dms: » ou « aws: ». La chaîne peut uniquement contenir l'ensembl e de lettres, de chiffres et d'espaces Unicode, '_', '.', '/', '=', '+', '-' (regex Java : `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`). Les clés doivent être propres à un ensemble de balises. Par exemple, vous ne pouvez pas avoir une paire-clé dans un ensemble de balises avec la clé identique mais des valeurs différentes comme projet/Trinity et projet/Xanadu.
Valeur	Une valeur est la valeur facultative de la balise. La valeur de la chaîne peut comporter de 1 à 256 caractères Unicode et ne peut pas être précédée de « dms: » ou « aws: ». La chaîne peut uniquement contenir l'ensemble de lettres, de chiffres et d'espaces Unicode, '_', '.', '/', '=', '+', '-' (regex Java : `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`). Les valeurs comprises dans un ensemble de balises ne doivent pas nécessairement être uniques et peuvent être null. Par exemple, vous pouvez avoir une paire clé-valeur dans un ensemble de balises appelé projet/Trinity et centre-de-coûts/Trinity.

Sécurité dans AWS Database Migration Service

Chez AWS, la sécurité dans le cloud est la priorité numéro 1. En tant que client AWS, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des organisations les plus pointilleuses en termes de sécurité.

La sécurité est une responsabilité partagée entre AWS et vous-même. Le modèle de responsabilité partagée décrit cette notion par les termes sécurité du cloud et sécurité dans le cloud :

- Sécurité du cloud : AWS est responsable de la protection de l'infrastructure qui exécute des services AWS dans le cloud AWS. AWS vous fournit également les services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de conformité AWS. Pour de plus amples informations sur les programmes de conformité qui s'appliquent à AWS DMS, veuillez consulter Services AWS concernés par le programme de conformité.
- Sécurité dans le cloud – Votre responsabilité est déterminée par le service AWS que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre organisation, et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de AWS DMS. Les rubriques suivantes expliquent comment configurer AWS DMS pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres services AWS pour surveiller et sécuriser vos ressources AWS DMS.

Vous pouvez gérer l'accès à vos ressources AWS DMS et à vos bases de données. La méthode que vous utilisez pour gérer l'accès dépend de la tâche de réplication que vous devez effectuer avec AWS DMS:

- Utilisez des politiques AWS Identity and Access Management (IAM) pour attribuer des autorisations qui déterminent qui est autorisé à gérer les ressources AWS DMS. AWS DMS exige que vous disposiez des autorisations appropriées si vous vous connectez en tant qu'utilisateur IAM. Par exemple, vous pouvez utiliser IAM pour déterminer qui est autorisé à créer, décrire, modifier et supprimer des instances de base de données et des clusters de bases de données, attribuer des balises à des ressources ou modifier des groupes de sécurité. Pour plus d'informations sur IAM et son utilisation avec AWS DMS, consultez Gestion des identités et des accès pour AWS Database Migration Service.

- AWS DMS utilise Secure Sockets Layer (SSL) pour vos connexions de point de terminaison avec Transport Layer Security (TLS). Pour de plus amples informations sur l'utilisation de SSL/TLS avec AWS DMS, veuillez consulter Utilisation du protocole SSL avec AWS Database Migration Service.
- AWS DMS utilise des clés de chiffrement AWS Key Management Service (AWS KMS) pour chiffrer le stockage utilisé par votre instance de réplication et ses informations de connexion au point de terminaison. AWS DMS utilise également des clés de chiffrement AWS KMS pour sécuriser les données cibles au repos pour les points de terminaison cibles Amazon S3 et Amazon Redshift. Pour de plus amples informations, veuillez consulter Définition d'une clé de chiffrement et spécification des autorisations AWS KMS.
- AWS DMS crée toujours votre instance de réplication dans un cloud privé virtuel (VPC) basé sur le service Amazon VPC pour optimiser le contrôle d'accès réseau. Pour les instances de base de données et les clusters d'instances, utilisez le même VPC que votre instance de réplication ou des VPC supplémentaires pour atteindre ce niveau de contrôle d'accès. Chaque réseau Amazon VPC que vous utilisez doit être associé à un groupe de sécurité dont les règles permettent à tout le trafic sur tous les ports de quitter (trafic sortant) le VPC. Cette approche permet la communication de l'instance de réplication avec vos points de terminaison de base de données source et cible, tant que les bonnes règles de trafic entrant sont activées sur ces points de terminaison.

  Pour plus d'informations sur les configurations réseau disponibles pour AWS DMS, consultez Configuration d'un réseau pour une instance de réplication. Pour plus d'informations sur la création d'une instance de base de données ou d'un cluster d'instances dans un VPC, consultez la documentation sur la sécurité et la gestion des clusters de vos bases de données Amazon dans la documentation AWS. Pour plus d'informations sur les configurations réseau prises en charge par AWS DMS, consultez Configuration d'un réseau pour une instance de réplication.
- Pour consulter les journaux de migration de base de données, vous devez détenir les autorisations appropriées d'Amazon CloudWatch Logs pour le rôle IAM que vous utilisez. Pour plus d'informations sur la journalisation pour AWS DMS, consultez Surveillance des tâches de réplication à l'aide d'Amazon CloudWatch.

Rubriques

- Protection des données dans AWS Database Migration Service
- Gestion des identités et des accès pour AWS Database Migration Service
- Validation de la conformité pour AWS Database Migration Service
- Résilience dans AWS Database Migration Service
- Sécurité de l'infrastructure dans AWS Database Migration Service

- Contrôle précis des accès à l'aide des noms de ressources et des balises
- Définition d'une clé de chiffrement et spécification des autorisations AWS KMS
- Sécurité de réseau pour AWS Database Migration Service
- Utilisation du protocole SSL avec AWS Database Migration Service
- Modification du mot de passe de base de données

Protection des données dans AWS Database Migration Service

Chiffrement des données

Vous pouvez activer le chiffrement des ressources de données des points de terminaison AWS DMS cibles pris en charge. AWS DMS chiffre également les connexions à AWS DMS et entre tous ses points de terminaison AWS DMS source et cible. En outre, vous pouvez gérer les clés utilisées par AWS DMS et les points de terminaison cibles pris en charge pour activer ce chiffrement.

Rubriques

- Chiffrement au repos
- Chiffrement en transit
- Gestion des clés

Chiffrement au repos

AWS DMS prend en charge le chiffrement au repos en vous permettant de spécifier le mode de chiffrement côté serveur que vous souhaitez utiliser pour envoyer (push) les données répliquées à Amazon S3 avant leur copie vers les points de terminaison cibles AWS DMS pris en charge. Pour spécifier ce mode de chiffrement, définissez l'attribut de connexion supplémentaire `encryptionMode` pour le point de terminaison. Si ce paramètre `encryptionMode` spécifie le mode de chiffrement des clés KMS, vous pouvez également créer des clés AWS KMS personnalisées spécifiquement pour chiffrer les données cibles pour les points de terminaison cibles AWS DMS suivants :

- Amazon Redshift : pour plus d'informations sur le paramètre `encryptionMode`, consultez Paramètres de point de terminaison lors de l'utilisation d'Amazon Redshift en tant que cible pour AWS DMS. Pour plus d'informations sur la création d'une clé de chiffrement AWS KMS personnalisée, consultez Création et utilisation de clés AWS KMS pour chiffrer les données cibles Amazon Redshift.
- Amazon S3 : pour plus d'informations sur le paramètre `encryptionMode`, consultez Paramètres de point de terminaison lors de l'utilisation d'Amazon S3 en tant que cible pour AWS DMS. Pour plus d'informations sur la création d'une clé de chiffrement AWS KMS personnalisée, consultez Création de clés AWS KMS pour chiffrer des objets cibles Amazon S3.

Chiffrement en transit

AWS DMS prend en charge le chiffrement en transit et veille ainsi à ce que les données qu'il réplique se déplacent en toute sécurité du point de terminaison source vers le point de terminaison cible. Cela inclut le chiffrement d'un compartiment S3 sur l'instance de réplication utilisée par votre tâche de réplication pour le stockage intermédiaire lorsque les données se déplacent dans le pipeline de réplication. Pour chiffrer les connexions de tâches aux points de terminaison sources et cibles, AWS DMS utilise le protocole SSL (Secure Socket Layer) ou le protocole TLS (Transport Layer Security). En chiffrant les connexions aux deux points de terminaison, AWS DMS garantit la sécurisation de vos données lorsqu'elles se déplacent à la fois du point de terminaison source vers la tâche de réplication et de la tâche de réplication vers le point de terminaison cible. Pour de plus amples informations sur l'utilisation de SSL/TLS avec AWS DMS, consultez Utilisation du protocole SSL avec AWS Database Migration Service.

AWS DMS prend en charge les clés par défaut et les clés personnalisées pour chiffrer à la fois le stockage de réplication intermédiaire et les informations de connexion. La gestion de ces clés s'effectue via AWS KMS. Pour de plus amples informations, veuillez consulter Définition d'une clé de chiffrement et spécification des autorisations AWS KMS.

Gestion des clés

AWS DMS prend en charge les clés par défaut ou les clés personnalisées pour chiffrer le stockage de réplication, les informations de connexion et le stockage de données cible pour certains points de terminaison cibles. La gestion de ces clés s'effectue via AWS KMS. Pour de plus amples informations, veuillez consulter Définition d'une clé de chiffrement et spécification des autorisations AWS KMS.

Confidentialité du trafic inter-réseau

Les connexions bénéficient d'une protection entre les points de terminaison AWS DMS source et cible dans la même région AWS, qu'elles s'exécutent sur site ou dans le cadre d'un service AWS dans le cloud. Notez qu'au moins un point de terminaison, une source ou une cible doit être exécuté dans le cadre d'un service AWS dans le cloud. Cette protection s'applique que ces composants partagent le même Virtual Private Cloud (VPC) ou qu'ils existent dans des VPC distincts, dans la mesure où les VPC se trouvent tous dans la même région AWS. Pour plus d'informations sur les configurations réseau prises en charge pour AWS DMS, consultez Configuration d'un réseau pour une instance de réplication. Pour plus d'informations sur les considérations de sécurité lors de l'utilisation de ces configurations réseau, consultez Sécurité de réseau pour AWS Database Migration Service.

Protection des données dans DMS Fleet Advisor

DMS Fleet Advisor collecte et analyse vos métadonnées de base de données afin de déterminer la taille appropriée de la cible de migration. DMS Fleet Advisor n'accède pas aux données des tables et ne les transfère pas. De plus, DMS Fleet Advisor ne suit pas l'utilisation des fonctionnalités de base de données et n'accède pas à vos statistiques d'utilisation.

Vous contrôlez l'accès à vos bases de données lorsque vous créez des utilisateurs de base de données que DMS Fleet Advisor utilise pour travailler avec vos bases de données. Vous accordez les privilèges requis à ces utilisateurs. Pour utiliser DMS Fleet Advisor, vous devez accorder à vos utilisateurs de base de données des autorisations de lecture. DMS Fleet Advisor ne modifie pas vos bases de données et ne nécessite aucune autorisation d'écriture. Pour de plus amples informations, veuillez consulter Création d'utilisateurs de base de données pour AWS DMS Fleet Advisor.

Vous pouvez utiliser le chiffrement des données dans vos bases de données. AWS DMS chiffre également les connexions au sein de DMS Fleet Advisor et de ses collecteurs de données.

Le collecteur de données DMS utilise l'API de protection des données (DPAPI) pour chiffrer, protéger et stocker les informations relatives à l'environnement du client et aux informations d'identification de base de données. DMS Fleet Advisor stocke ces données chiffrées dans un fichier sur le serveur où fonctionne votre collecteur de données DMS. DMS Fleet Advisor ne transfère pas ces données depuis ce serveur. Pour plus d'informations sur DPAPI, consultez Procédure : utiliser la protection des données.

Après avoir installé le collecteur de données DMS, vous pouvez consulter toutes les requêtes que cette application exécute pour collecter les métriques. Vous pouvez exécuter le collecteur de données DMS en mode hors connexion, puis passer en revue les données collectées sur votre serveur. Vous pouvez également passer en revue ces données collectées dans votre compartiment Amazon S3. Pour de plus amples informations, veuillez consulter Comment fonctionne le collecteur de données DMS ?.

Gestion des identités et des accès pour AWS Database Migration Service

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser AWS DMS les ressources. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Public ciblé
- Authentification par des identités
- Gestion des accès à l'aide de politiques
- Comment AWS Database Migration Service fonctionne avec IAM
- AWS Database Migration Service exemples de politiques basées sur l'identité
- Exemples de politiques basées sur les ressources pour AWS KMS
- Utilisation de secrets pour accéder aux points de terminaison AWS Database Migration Service
- Utilisation des rôles liés aux services pour AWS DMS
- Résolution des problèmes AWS Database Migration Service d'identité et d'accès
- Autorisations IAM nécessaires pour utiliser AWS DMS
- Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS
- Prévention du problème de l'adjoint confus entre services
- Politiques AWS gérées pour AWS Database Migration Service

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez. AWS DMS

Utilisateur du service : si vous utilisez le AWS DMS service pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de nouvelles AWS DMS fonctionnalités pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne

pouvez pas accéder à une fonctionnalité dans AWS DMS, consultez Résolution des problèmes AWS Database Migration Service d'identité et d'accès.

Administrateur du service — Si vous êtes responsable des AWS DMS ressources de votre entreprise, vous avez probablement un accès complet à AWS DMS. C'est à vous de déterminer les AWS DMS fonctionnalités et les ressources auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM avec AWS DMS, voirComment AWS Database Migration Service fonctionne avec IAM.

Administrateur IAM : si vous êtes un administrateur IAM, vous souhaiterez peut-être en savoir plus sur la façon d'écrire des politiques pour gérer l'accès à AWS DMS. Pour consulter des exemples de politiques AWS DMS basées sur l'identité que vous pouvez utiliser dans IAM, consultez. AWS Database Migration Service exemples de politiques basées sur l'identité

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour

signer vous-même les demandes, consultez la section Signature des demandes AWS d'API dans le guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, veuillez consulter Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Utilisation de l'authentification multifactorielle (MFA) dans l'interface AWS dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur root pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur root et utilisez-les pour effectuer les tâches que seul l'utilisateur root peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur root, consultez Tâches nécessitant les informations d'identification de l'utilisateur root dans le Guide de l'utilisateur IAM.

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez la section Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification du Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez Quand créer un utilisateur IAM (au lieu d'un rôle) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité au sein de vous Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un rôle IAM dans le en AWS Management Console changeant de rôle. Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez la section Utilisation de rôles IAM dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez la section Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, veuillez consulter la rubrique Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service

exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, une fonction de service ou un rôle lié au service.

- Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, l'action que vous effectuez est susceptible de lancer une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez Sessions de transmission d'accès.
- Fonction du service : il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.
- Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes d'API. AWS CLI AWS Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez la section Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2 dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez la section Quand créer un rôle IAM (au lieu d'un utilisateur) dans le Guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez Présentation des politiques JSON dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations relatives à une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez la section Création de politiques IAM dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre

une politique gérée et une politique en ligne, consultez la section Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifient quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez Présentation des listes de contrôle d'accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d'autorisations : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations qui en résultent représentent la combinaison des politiques basées sur

l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.

- Politiques de contrôle des services (SCP) — Les SCP sont des politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans. AWS Organizations AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les organisations et les SCP, consultez Fonctionnement des SCP dans le Guide de l'utilisateur AWS Organizations .
- politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de la séance obtenue sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations, consultez Politiques de session dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations obtenues sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment AWS Database Migration Service fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à AWS DMS, vous devez connaître les fonctionnalités IAM disponibles. AWS DMS Pour obtenir une vue d'ensemble de la façon dont AWS DMS les autres AWS services fonctionnent avec IAM, consultez la section AWS Services compatibles avec IAM dans le Guide de l'utilisateur d'IAM.

Rubriques

- AWS DMS Politiques basées sur l'identité
- AWS DMS Politiques basées sur les ressources
- Autorisation basée sur les balises AWS DMS
- Rôles IAM pour AWS DMS
- Gestion des identités et des accès pour DMS Fleet Advisor

AWS DMS Politiques basées sur l'identité

Avec les politiques basées sur l'identité IAM, vous pouvez spécifier les actions et les ressources autorisées ou refusées, ainsi que les conditions selon lesquelles les actions sont autorisées ou refusées. AWS DMS prend en charge des actions, des ressources et des clés de condition spécifiques. Pour en savoir plus sur tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Actions

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Les actions de politique en AWS DMS cours utilisent le préfixe suivant avant l'action :`dms:`. Par exemple, pour autoriser quelqu'un à créer une tâche de réplication avec l'opération d' AWS DMS `CreateReplicationTask`API, vous devez inclure l'`dms:CreateReplicationTask`action dans sa politique. Les déclarations de politique doivent inclure un `NotAction` élément `Action` ou. AWS DMS définit son propre ensemble d'actions décrivant les tâches que vous pouvez effectuer avec ce service.

Pour spécifier plusieurs actions dans une seule instruction, séparez-les par des virgules, comme suit :

```
"Action": [
      "dms:action1",
      "dms:action2"
```

Vous pouvez aussi préciser plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante.

```
"Action": "dms:Describe*"
```

Pour consulter la liste des AWS DMS actions, reportez-vous à la section Actions définies par AWS Database Migration Service dans le guide de l'utilisateur IAM.

Ressources

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets pour lesquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

AWS DMS fonctionne avec les ressources suivantes :

- Certificats
- Points de terminaison

- Abonnements aux événements
- Instances de réplication
- Groupes de sous-réseaux de réplication (sécurité)
- Tâches de réplication

La ou les ressources AWS DMS requises dépendent de l'action ou des actions que vous invoquez. Vous avez besoin d'une stratégie qui autorise ces actions au niveau de la ou des ressources associées spécifiées par les ARN correspondants.

Par exemple, une ressource de point de AWS DMS terminaison possède l'ARN suivant :

```
arn:${Partition}:dms:${Region}:${Account}:endpoint/${InstanceId}
```

Pour plus d'informations sur le format des ARN, consultez Amazon Resource Names (ARN) et espaces de noms de AWS services.

Par exemple, pour spécifier l'instance de point de terminaison `1A2B3C4D5E6F7G8H9I0J1K2L3M` de la région `use-east-2` dans votre instruction, utilisez l'ARN suivant.

```
"Resource": "arn:aws:dms:us-east-2:987654321098:endpoint/1A2B3C4D5E6F7G8H9I0J1K2L3M"
```

Pour spécifier tous les points de terminaison qui appartiennent à un compte spécifique, utilisez le caractère générique (*).

```
"Resource": "arn:aws:dms:us-east-2:987654321098:endpoint/*"
```

Certaines AWS DMS actions, telles que celles relatives à la création de ressources, ne peuvent pas être effectuées sur une ressource spécifique. Dans ces cas-là, vous devez utiliser le caractère générique (*).

```
"Resource": "*"
```

Certaines actions AWS DMS d'API impliquent plusieurs ressources. Par exemple, `StartReplicationTask` démarre une tâche de réplication et la connecte à deux ressources de point de terminaison de base de données (une source et une cible). Un utilisateur IAM doit donc disposer des autorisations nécessaires pour lire le point de terminaison source et écrire sur le point

de terminaison cible. Pour spécifier plusieurs ressources dans une seule instruction, séparez leurs ARN par des virgules.

```
"Resource": [
      "resource1",
      "resource2" ]
```

Pour plus d'informations sur le contrôle de l'accès aux AWS DMS ressources à l'aide de politiques, consultezUtilisation des noms de ressources pour contrôler l'accès. Pour afficher une liste des types de ressources AWS DMS et de leurs ARN, consultez Types de ressources définis par AWS Database Migration Service dans le Guide de l'utilisateur IAM. Pour savoir grâce à quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par AWS Database Migration Service.

Clés de condition

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

AWS DMS définit son propre ensemble de clés de condition et prend également en charge l'utilisation de certaines clés de condition globales. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

AWS DMS définit un ensemble de balises standard que vous pouvez utiliser dans ses clés de condition et vous permet également de définir vos propres balises personnalisées. Pour de plus amples informations, veuillez consulter Utilisation de balises pour contrôler l'accès.

Pour consulter la liste des clés de AWS DMS condition, reportez-vous à la section Clés de AWS Database Migration Service condition du guide de l'utilisateur IAM. Pour découvrir les actions et les ressources avec lesquelles vous pouvez utiliser une clé de condition, consultez Actions définies par AWS Database Migration Service et Ressources définies par AWS Database Migration Service.

Exemples

Pour consulter des exemples de politiques AWS DMS basées sur l'identité, consultez. AWS Database Migration Service exemples de politiques basées sur l'identité

AWS DMS Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON qui spécifient les actions qu'un principal spécifié peut effectuer sur une AWS DMS ressource donnée et dans quelles conditions. AWS DMS prend en charge les politiques d'autorisation basées sur les ressources pour les clés de AWS KMS chiffrement que vous créez pour chiffrer les données migrées vers les points de terminaison cibles pris en charge. Les points de terminaison cibles pris en charge incluent Amazon Redshift et Amazon S3. Grâce aux stratégies basées sur les ressources, vous pouvez accorder l'autorisation d'utiliser ces clés de chiffrement à d'autres comptes pour chaque point de terminaison cible.

Pour permettre un accès comptes multiples , vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une stratégie basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource se trouvent dans des AWS comptes différents, vous devez également accorder à l'entité principale l'autorisation d'accéder à la ressource. Accordez l'autorisation en attachant une stratégie basée sur les identités à l'entité. Toutefois, si une stratégie basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre stratégie basée sur l'identité n'est requise. Pour plus d'informations,

consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

Le AWS DMS service ne prend en charge qu'un seul type de stratégie basée sur les ressources, appelée politique clé, qui est attachée à une clé de AWS KMS chiffrement. Cette stratégie définit les entités principales (comptes, utilisateurs, rôles et utilisateurs fédérés) qui peuvent chiffrer les données migrées sur le point de terminaison cible pris en charge.

Pour découvrir comment associer une stratégie basée sur les ressources à une clé de chiffrement que vous créez pour les points de terminaison cibles pris en charge, consultez Création et utilisation de clés AWS KMS pour chiffrer les données cibles Amazon Redshift et Création de clés AWS KMS pour chiffrer des objets cibles Amazon S3.

Exemples

Pour des exemples de politiques AWS DMS basées sur les ressources, voir. Exemples de politiques basées sur les ressources pour AWS KMS

Autorisation basée sur les balises AWS DMS

Vous pouvez associer des balises aux AWS DMS ressources ou transmettre des balises dans une demande à AWS DMS. Pour contrôler l'accès en fonction des balises, vous devez fournir les informations relatives aux balises dans l'élément de condition d'une politique à l'aide de la clé de `aws:TagKeys` condition `dms:ResourceTag/`*`key-name`*`aws:RequestTag/`*`key-name`*, ou. AWS DMS définit un ensemble de balises standard que vous pouvez utiliser dans ses clés de condition et vous permet également de définir vos propres balises personnalisées. Pour de plus amples informations, veuillez consulter Utilisation de balises pour contrôler l'accès.

Pour obtenir un exemple de stratégie basée sur l'identité, qui limite l'accès à une ressource en fonction des balises, consultez Accès aux ressources AWS DMS en fonction des balises.

Rôles IAM pour AWS DMS

Un rôle IAM est une entité de votre AWS compte qui possède des autorisations spécifiques.

Utilisation d'informations d'identification temporaires avec AWS DMS

Vous pouvez utiliser des informations d'identification temporaires pour vous connecter à l'aide de la fédération, endosser un rôle IAM ou encore pour endosser un rôle intercompte. Vous obtenez des informations d'identification de sécurité temporaires en appelant des opérations d' AWS STS API telles que AssumeRoleou GetFederationToken.

AWS DMS prend en charge l'utilisation d'informations d'identification temporaires.

Rôles liés à un service

Les rôles liés aux AWS services permettent aux services d'accéder aux ressources d'autres services pour effectuer une action en votre nom. Les rôles liés à un service s'affichent dans votre compte IAM et sont la propriété du service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus de détails sur la création ou la gestion des rôles AWS DMS liés à un service, consultez. Utilisation des rôles liés aux services

Rôles de service

Cette fonction permet à un service d'endosser une fonction du service en votre nom. Ce rôle autorise le service à accéder à des ressources d'autres services pour effectuer une action en votre nom. Les fonctions du service s'affichent dans votre compte IAM et sont la propriété du compte. Cela signifie qu'un administrateur IAM peut modifier les autorisations associées à ce rôle. Toutefois, une telle action peut perturber le bon fonctionnement du service.

AWS DMS prend en charge deux types de rôles de service que vous devez créer pour utiliser certains points de terminaison source ou cible :

- Rôles autorisés à autoriser l'accès du AWS DMS aux points de terminaison source et cible suivants (ou à leurs ressources) :
  - Amazon DynamoDB en tant que cible : pour plus d'informations, consultez Prérequis pour l'utilisation de DynamoDB en tant que cible pour AWS Database Migration Service.
  - OpenSearch comme cible — Pour plus d'informations, voirConditions préalables à l'utilisation d'Amazon OpenSearch Service en tant que cible pour AWS Database Migration Service.
  - Amazon Kinesis en tant que cible : pour plus d'informations, consultez Conditions préalables à l'utilisation d'un flux de données Kinesis comme cible pour AWS Database Migration Service.
  - Amazon Redshift en tant que cible : vous devez créer le rôle spécifié uniquement pour créer une clé de chiffrement KMS personnalisée permettant de chiffrer les données cibles ou pour définir un compartiment S3 personnalisé dans lequel stocker les tâches intermédiaires. Pour plus d'informations, consultez Création et utilisation de clés AWS KMS pour chiffrer les données cibles Amazon Redshift ou Paramètres du compartiment Amazon S3.
  - Amazon S3 en tant que source ou cible : pour plus d'informations, consultez Conditions préalables lors de l'utilisation d'Amazon S3 en tant que source pour AWS DMS ou Prérequis pour l'utilisation d'Amazon S3 en tant que cible.

Par exemple, pour lire des données à partir d'un point de terminaison source S3 ou pour transférer des données vers un point de terminaison cible S3, vous devez créer un rôle de service comme condition préalable à l'accès à S3 pour chacune de ces opérations.

- Rôles dotés des autorisations requises pour utiliser l'API AWS CLI et AWS DMS — Les deux rôles IAM que vous devez créer sont `dms-vpc-role` et. `dms-cloudwatch-logs-role` Si vous utilisez Amazon Redshift comme base de données cible, vous devez également créer et ajouter le rôle IAM `dms-access-for-endpoint` à votre compte. AWS Pour de plus amples informations, veuillez consulter Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS.

Choisir un rôle IAM dans AWS DMS

Si vous utilisez l'API AWS CLI ou l'API AWS DMS pour la migration de votre base de données, vous devez ajouter certains rôles IAM à votre AWS compte avant de pouvoir utiliser les fonctionnalités de AWS DMS. Deux d'entre eux sont `dms-vpc-role` et `dms-cloudwatch-logs-role`. Si vous utilisez Amazon Redshift comme base de données cible, vous devez également ajouter le rôle IAM `dms-access-for-endpoint` à votre compte. AWS Pour de plus amples informations, veuillez consulter Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS.

Gestion des identités et des accès pour DMS Fleet Advisor

Avec les politiques basées sur l'identité IAM, vous pouvez spécifier des actions et des ressources autorisées ou refusées ainsi que les conditions selon lesquelles les actions sont autorisées ou refusées. DMS Fleet Advisor prend en charge des actions, ressources et clés de condition spécifiques. Pour en savoir plus sur tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

DMS Fleet Advisor utilise des rôles IAM pour accéder à Amazon Simple Storage Service. Un rôle IAM est une entité de votre AWS compte qui possède des autorisations spécifiques. Pour de plus amples informations, veuillez consulter Création de ressources IAM.

AWS Database Migration Service exemples de politiques basées sur l'identité

Par défaut, les utilisateurs et les rôles IAM ne sont pas autorisés à créer ou modifier les ressources AWS DMS . Ils ne peuvent pas non plus effectuer de tâches à l'aide de l' AWS API AWS Management Console AWS CLI, ou. Un administrateur IAM doit créer des politiques IAM autorisant les utilisateurs et les rôles à exécuter des opérations d'API spécifiques sur les ressources spécifiées

dont ils ont besoin. Il doit ensuite attacher ces politiques aux utilisateurs ou aux groupes IAM ayant besoin de ces autorisations.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, veuillez consulter Création de politiques dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la console AWS DMS
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations
- Accès à un compartiment Amazon S3
- Accès aux ressources AWS DMS en fonction des balises

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer AWS DMS des ressources dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez Politiques gérées par AWS ou Politiques gérées par AWS pour les fonctions professionnelles dans le Guide de l'utilisateur IAM.
- Accorder des autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, n'accordez que les autorisations nécessaires à l'exécution de la tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation de IAM pour appliquer des autorisations, consultez Politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par

exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour de plus amples informations, consultez Validation de politique IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour de plus amples informations, consultez Configuration de l'accès aux API protégé par MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Utilisation de la console AWS DMS

La politique suivante vous donne accès au AWS DMS, y compris à la console AWS DMS, et spécifie également les autorisations pour certaines actions requises par d'autres services Amazon tels qu'Amazon EC2.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "dms:*",
            "Resource": "arn:aws:dms:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "kms:ListAliases",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole",
                "iam:CreateRole",
                "iam:AttachRolePolicy"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:Get*",
                "cloudwatch:List*"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:FilterLogEvents",
                "logs:GetLogEvents"
```

```
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
    ]
}
```

Une description de ces autorisations peut vous aider à mieux comprendre pourquoi chacune d'entre elles est nécessaire à l'utilisation de la console.

La section suivante est obligatoire pour permettre à l'utilisateur d'afficher ses clés AWS KMS disponibles et leurs alias sur la console. Cette entrée n'est pas requise si vous connaissez l'Amazon Resource Name (ARN) pour la clé KMS et que vous utilisez uniquement l' AWS Command Line Interface (AWS CLI).

```
{
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

La section suivante est requise lorsque certains types de points de terminaison nécessitent la transmission d'un rôle ARN avec le point de terminaison. En outre, si les AWS DMS rôles requis ne sont pas créés à l'avance, la AWS DMS console est en mesure de créer le rôle. Si tous les rôles sont configurés à l'avance, ils sont tous nécessaires dans `iam:GetRole` et `iam:PassRole`. Pour plus d'informations sur les rôles , consultez Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS.

```
{
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole",
                "iam:CreateRole",
                "iam:AttachRolePolicy"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
```

```
        }
```

La section suivante est obligatoire car elle AWS DMS doit créer l'instance Amazon EC2 et configurer le réseau pour l'instance de réplication créée. Comme ces ressources existent dans le compte du client, il est obligatoire de pouvoir effectuer ces actions au nom du client.

```
{
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

La section suivante est obligatoire pour permettre à l'utilisateur d'afficher les métriques de l'instance de réplication.

```
{
            "Effect": "Allow",
            "Action": [
                "cloudwatch:Get*",
                "cloudwatch:List*"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

Cette section est obligatoire pour permettre à l'utilisateur d'afficher les journaux de réplication.

```
{
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:FilterLogEvents",
```

```
                "logs:GetLogEvents"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

La console AWS DMS crée plusieurs rôles qui sont automatiquement associés à votre AWS compte lorsque vous utilisez la console AWS DMS. Si vous utilisez l'API AWS Command Line Interface (AWS CLI) ou AWS DMS pour votre migration, vous devez ajouter ces rôles à votre compte. Pour plus d'informations sur l'ajout de ces rôles, consultez Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS.

Pour plus d'informations sur les exigences relatives à l'utilisation de cette politique pour accéder au AWS DMS, consultezAutorisations IAM nécessaires pour utiliser AWS DMS.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
```

```
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Accès à un compartiment Amazon S3

AWS DMS utilise des compartiments Amazon S3 comme stockage intermédiaire pour la migration de bases de données. En général, AWS DMS gère les compartiments S3 par défaut à cette fin. Toutefois, dans certains cas, en particulier lorsque vous utilisez l'API AWS CLI ou l'API AWS DMS, AWS DMS vous permet de spécifier votre propre compartiment S3 à la place. Par exemple, vous pouvez spécifier votre propre compartiment S3 pour la migration de données vers un point de terminaison cible Amazon Redshift. Dans ce cas, vous devez créer un rôle avec des autorisations basées sur la `AmazonDMSRedshiftS3Role` politique AWS gérée.

L'exemple suivant présente une version de la politique `AmazonDMSRedshiftS3Role`. Cela permet à AWS DMS d'accorder à un utilisateur IAM de votre AWS compte l'accès à l'un de vos compartiments Amazon S3. Il permet également à l'utilisateur d'ajouter, de mettre à jour et de supprimer des objets.

En plus de l'octroi des autorisations `s3:PutObject`, `s3:GetObject` et `s3:DeleteObject` à l'utilisateur, la stratégie octroie aussi les autorisations `s3:ListAllMyBuckets`, `s3:GetBucketLocation` et `s3:ListBucket`. Ces conditions supplémentaires sont requises par la console. D'autres autorisations permettent à AWS DMS de gérer le cycle de vie du bucket. En outre, l'action `s3:GetObjectAcl` est nécessaire pour pouvoir copier des objets.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:ListBucket",
                "s3:DeleteBucket",
```

```
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "s3:GetObjectVersion",
                "s3:GetBucketPolicy",
                "s3:PutBucketPolicy",
                "s3:GetBucketAcl",
                "s3:PutBucketVersioning",
                "s3:GetBucketVersioning",
                "s3:PutLifecycleConfiguration",
                "s3:GetLifecycleConfiguration",
                "s3:DeleteBucketPolicy"
            ],
            "Resource": "arn:aws:s3:::dms-*"
        }
    ]
}
```

Pour plus d'informations sur la création d'un rôle basé sur cette stratégie, consultez Paramètres du compartiment Amazon S3.

Accès aux ressources AWS DMS en fonction des balises

Vous pouvez utiliser des conditions dans votre politique basée sur l'identité pour contrôler l'accès aux ressources AWS DMS en fonction des balises. Cet exemple montre comment créer une politique qui autorise l'accès à tous les points de terminaison AWS DMS. Toutefois, l'autorisation est accordée uniquement si la balise `Owner` contient le nom d'utilisateur approprié.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "dms:*",
            "Resource": "arn:aws:dms:*:*:endpoint/*",
            "Condition": {
                "StringEquals": {"dms:endpoint-tag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

Vous pouvez rattacher cette politique aux utilisateurs IAM de votre compte. Si un utilisateur nommé `richard-roe` tente d'accéder à un AWS DMS point de terminaison, la base de données de points de terminaison doit être `0wner=richard-roe` étiquetée ou`owner=richard-roe`. Dans le cas contraire, l'utilisateur se voit refuser l'accès. La clé de condition d'étiquette `0wner` correspond à la fois à `0wner` et à `owner`, car les noms de clé de condition ne sont pas sensibles à la casse. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur les ressources pour AWS KMS

AWS DMS vous permet de créer des clés de AWS KMS chiffrement personnalisées pour chiffrer les données des terminaux cibles pris en charge. Pour découvrir comment créer et attacher une stratégie de clé à la clé de chiffrement que vous générez pour le chiffrement des données cibles prises en charge, consultez Création et utilisation de clés AWS KMS pour chiffrer les données cibles Amazon Redshift et Création de clés AWS KMS pour chiffrer des objets cibles Amazon S3.

Rubriques

- Une politique pour une clé de AWS KMS chiffrement personnalisée afin de chiffrer les données cibles d'Amazon Redshift
- Une politique pour une clé de AWS KMS chiffrement personnalisée afin de chiffrer les données cibles d'Amazon S3

Une politique pour une clé de AWS KMS chiffrement personnalisée afin de chiffrer les données cibles d'Amazon Redshift

L'exemple suivant présente le code JSON de la stratégie de clé créée pour une clé de chiffrement AWS KMS permettant de chiffrer les données cibles Amazon Redshift.

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:root"
        ]
```

```
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow access for Key Administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/Admin"
        ]
      },
      "Action": [
        "kms:Create*",
        "kms:Describe*",
        "kms:Enable*",
        "kms:List*",
        "kms:Put*",
        "kms:Update*",
        "kms:Revoke*",
        "kms:Disable*",
        "kms:Get*",
        "kms:Delete*",
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ScheduleKeyDeletion",
        "kms:CancelKeyDeletion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/DMS-Redshift-endpoint-access-role"
        ]
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
```

```
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow attachment of persistent resources",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/DMS-Redshift-endpoint-access-role"
        ]
      },
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": "*",
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": true
        }
      }
    }
  ]
}
```

Ici, vous pouvez voir à quel endroit la stratégie de clé référence le rôle permettant d'accéder aux données du point de terminaison cible Amazon Redshift que vous avez créées avant de créer la clé. Dans cet exemple, il s'agit de `DMS-Redshift-endpoint-access-role`. Vous pouvez également voir les différentes actions clés autorisées pour les différentes entités (utilisateurs et rôles). Par exemple, tout utilisateur avec `DMS-Redshift-endpoint-access-role` peut chiffrer, déchiffrer et rechiffrer les données cibles. Un tel utilisateur peut également générer des clés de données à exporter afin de AWS KMS chiffrer les données externes. Ils peuvent également renvoyer des informations détaillées sur une AWS KMS clé, telles que la clé que vous venez de créer. En outre, un utilisateur avec ce rôle peut gérer les pièces jointes aux ressources AWS , telles que le point de terminaison cible.

Une politique pour une clé de AWS KMS chiffrement personnalisée afin de chiffrer les données cibles d'Amazon S3

L'exemple suivant présente le code JSON de la stratégie de clé créée pour une clé de chiffrement AWS KMS permettant de chiffrer les données cibles Amazon S3.

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:root"
        ]
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow access for Key Administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/Admin"
        ]
      },
      "Action": [
        "kms:Create*",
        "kms:Describe*",
        "kms:Enable*",
        "kms:List*",
        "kms:Put*",
        "kms:Update*",
        "kms:Revoke*",
        "kms:Disable*",
        "kms:Get*",
        "kms:Delete*",
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ScheduleKeyDeletion",
```

```
        "kms:CancelKeyDeletion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/DMS-S3-endpoint-access-role"
        ]
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow attachment of persistent resources",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/DMS-S3-endpoint-access-role"
        ]
      },
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": "*",
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": true
        }
      }
    }
  ]
```

Ici, vous pouvez voir à quel endroit la stratégie de clé référence le rôle permettant d'accéder aux données du point de terminaison cible Amazon S3 que vous avez créées avant de créer la clé. Dans cet exemple, il s'agit de `DMS-S3-endpoint-access-role`. Vous pouvez également voir les différentes actions clés autorisées pour les différentes entités (utilisateurs et rôles). Par exemple, tout utilisateur avec `DMS-S3-endpoint-access-role` peut chiffrer, déchiffrer et rechiffrer les données cibles. Un tel utilisateur peut également générer des clés de données à exporter afin de AWS KMS chiffrer les données externes. Ils peuvent également renvoyer des informations détaillées sur une AWS KMS clé, telles que la clé que vous venez de créer. En outre, un utilisateur avec ce rôle peut gérer les pièces jointes aux ressources AWS , telles que le point de terminaison cible.

Utilisation de secrets pour accéder aux points de terminaison AWS Database Migration Service

En effet AWS DMS, un secret est une clé cryptée que vous pouvez utiliser pour représenter un ensemble d'informations d'identification utilisateur afin d'authentifier, par le biais d'une authentification secrète, la connexion à la base de données pour un point de terminaison AWS DMS source ou cible pris en charge. Pour un point de terminaison Oracle qui utilise également Oracle Automatic Storage Management (ASM), AWS DMS nécessite un secret supplémentaire représentant les informations d'identification de l'utilisateur pour accéder à Oracle ASM.

Vous pouvez créer le ou les secrets qui AWS DMS nécessitent une authentification secrète à l'aide AWS Secrets Manager d'un service permettant de créer, de stocker et de récupérer en toute sécurité des informations d'identification pour accéder aux applications, aux services et aux ressources informatiques dans le cloud et sur site. Cela inclut la prise en charge de la rotation périodique automatique de la valeur de secret chiffrée sans votre intervention, offrant ainsi un niveau de sécurité supplémentaire pour vos informations d'identification. L'activation de la rotation des valeurs secrètes garantit AWS Secrets Manager également que cette rotation des valeurs secrètes s'effectue sans aucun effet sur la migration de base de données qui repose sur le secret. Pour authentifier secrètement une connexion à une base de données de point de terminaison, créez un secret dont vous affectez l'identité ou l'ARN à `SecretsManagerSecretId`, que vous incluez dans les paramètres du point de terminaison. Pour authentifier secrètement Oracle ASM dans le cadre d'un point de terminaison Oracle, créez un secret dont vous affectez l'identité ou l'ARN à `SecretsManagerOracleAsmSecretId`, que vous incluez également dans les paramètres du point de terminaison.

> **Note**
>
> Vous ne pouvez pas utiliser les informations d'identification principales gérées par Amazon RDS Aurora. Ces informations d'identification n'incluent pas les informations relatives à l'hôte ou au port, qui sont AWS DMS nécessaires pour établir des connexions. Créez plutôt un nouvel utilisateur et un nouveau secret. Pour en savoir plus sur la création d'un utilisateur et d'un secret, consultez la section Utilisation du AWS Management Console pour créer un rôle d'accès secret et secret ci-dessous.

Pour plus d'informations AWS Secrets Manager, voir Qu'est-ce que AWS Secrets Manager ? dans le guide de AWS Secrets Manager l'utilisateur.

AWS DMS prend en charge l'authentification secrète pour les bases de données sur site ou AWS gérées suivantes sur les points de terminaison source et cible pris en charge :

- Amazon DocumentDB
- IBM Db2 LUW
- Microsoft SQL Server
- MongoDB
- MySQL
- Oracle
- PostgreSQL
- Amazon Redshift
- SAP ASE

Pour vous connecter à l'une de ces bases de données, vous pouvez entrer l'un des ensembles de valeurs suivants, mais pas les deux, dans les paramètres du point de terminaison :

- Valeurs en texte clair pour authentifier la connexion à la base de données à l'aide des paramètres `UserName`, `Password`, `ServerName` et `Port`. Pour un point de terminaison Oracle qui utilise également Oracle ASM, incluez des valeurs en texte clair supplémentaires pour authentifier ASM à l'aide des paramètres `AsmUserName`, `AsmPassword` et `AsmServerName`.
- Authentification secrète à l'aide de valeurs pour les paramètres `SecretsManagerSecretId` et `SecretsManagerAccessRoleArn`. Pour un point de terminaison Oracle utilisant Oracle ASM, incluez des valeurs supplémentaires pour les paramètres

`SecretsManagerOracleAsmSecretId` et `SecretsManagerOracleAsmAccessRoleArn`. Les valeurs de secret de ces paramètres peuvent inclure les valeurs suivantes pour :

- `SecretsManagerSecretId` : Amazon Resource Name (ARN) complet, ARN partiel ou nom convivial d'un secret que vous avez créé dans AWS Secrets Manager pour accéder à la base de données du point de terminaison.
- `SecretsManagerAccessRoleArn`— L'ARN d'un rôle d'accès secret que vous avez créé dans IAM pour permettre AWS DMS l'accès à ce `SecretsManagerSecretId` secret en votre nom.
- `SecretsManagerOracleAsmSecretId` : Amazon Resource Name (ARN) complet, ARN partiel ou nom convivial d'un secret que vous avez créé dans AWS Secrets Manager pour accéder à Oracle ASM.
- `SecretsManagerOracleAsmAccessRoleArn` : ARN d'un rôle d'accès secret que vous avez créé dans IAM pour permettre à AWS DMS d'accéder en votre nom à ce secret `SecretsManagerOracleAsmSecretId`.

> **Note**
>
> Vous pouvez également utiliser un rôle d'accès secret unique pour fournir un AWS DMS accès à la fois au `SecretsManagerSecretId` secret et au `SecretsManagerOracleAsmSecretId` secret. Si vous créez ce rôle d'accès secret unique pour les deux secrets, assurez-vous d'affecter le même ARN pour ce rôle d'accès à `SecretsManagerAccessRoleArn` et `SecretsManagerOracleAsmAccessRoleArn`. Par exemple, si l'ARN de votre rôle d'accès secret pour les deux secrets est affecté à la variable `ARN2xsecrets`, vous pouvez définir ces paramètres d'ARN comme suit :
>
> ```
> SecretsManagerAccessRoleArn = ARN2xsecrets;
> SecretsManagerOracleAsmAccessRoleArn = ARN2xsecrets;
> ```

Pour plus d'informations sur la création de ces valeurs, consultez Utilisation du AWS Management Console pour créer un rôle d'accès secret et secret.

Après avoir créé et spécifié les paramètres de point de terminaison du secret et du rôle d'accès secret requis pour vos points de terminaison, mettez à jour les autorisations sur les comptes d'utilisateurs qui exécuteront la demande d'API `CreateEndpoint` ou `ModifyEndpoint` avec ces détails de secret. Assurez-vous que ces autorisations de compte incluent l'`IAM:GetRole`autorisation

sur le rôle d'accès secret et l'`SecretsManager:DescribeSecret`autorisation sur le secret. AWS DMS nécessite ces autorisations pour valider à la fois le rôle d'accès et son secret.

Pour fournir et vérifier les autorisations utilisateur requises

1. Connectez-vous à la AWS Identity and Access Management console AWS Management Console et ouvrez-la à l'adressehttps://console.aws.amazon.com/iam/.
2. Choisissez Utilisateurs, puis sélectionnez l'ID utilisateur utilisé pour effectuer des appels d'API `CreateEndpoint` et `ModifyEndpoint`.
3. Dans l'onglet Autorisations, choisissez {} JSON.
4. Assurez-vous que l'utilisateur dispose des autorisations suivantes.

```
{
 "Statement": [{
   "Effect": "Allow",
   "Action": [
    "iam:GetRole",
    "iam:PassRole"
   ],
   "Resource": "SECRET_ACCESS_ROLE_ARN"
  },
  {
   "Effect": "Allow",
   "Action": "secretsmanager:DescribeSecret",
   "Resource": "SECRET_ARN"
  }
 ]
}
```

5. Si l'utilisateur ne dispose pas de ces autorisations, ajoutez-les.
6. Si vous utilisez un rôle IAM pour effectuer des appels d'API DMS, répétez les étapes ci-dessus pour le rôle correspondant.
7. Ouvrez un terminal et utilisez le AWS CLI pour vérifier que les autorisations sont correctement accordées en assumant le rôle ou l'utilisateur utilisé ci-dessus.

   a. Validez l'autorisation de l'utilisateur SecretAccessRole à l'aide de la `get-role` commande IAM.

   ```
   aws iam get-role --role-name ROLE_NAME
   ```

Remplacez *ROLE_NAME* par le nom de `SecretsManagerAccessRole`.

Si la commande renvoie un message d'erreur, assurez-vous que les autorisations ont été correctement accordées.

b. Validez l'autorisation de l'utilisateur sur le secret à l'aide de la commande Secrets Manager `describe-secret`.

```
aws secretsmanager describe-secret --secret-id SECRET_NAME OR SECRET_ARN --
region=REGION_NAME
```

L'utilisateur peut être le nom convivial, l'ARN partiel ou l'ARN complet. Pour plus d'informations, consultez describe-secret.

Si la commande renvoie un message d'erreur, assurez-vous que les autorisations ont été correctement accordées.

Utilisation du AWS Management Console pour créer un rôle d'accès secret et secret

Vous pouvez utiliser le AWS Management Console pour créer un secret pour l'authentification des terminaux et pour créer la politique et le rôle AWS DMS permettant d'accéder au secret en votre nom.

Pour créer un secret à l'aide du AWS Management Console qui AWS DMS peut être utilisé pour authentifier une base de données pour les connexions aux points de terminaison source et cible

1. Connectez-vous à la AWS Secrets Manager console AWS Management Console et ouvrez-la à l'adressehttps://console.aws.amazon.com/secretsmanager/.
2. Choisissez Store a new secret (Stocker un nouveau secret).
3. Sous Sélectionner un type de secret sur la page Stocker un nouveau secret, choisissez Autre type de secrets, puis Texte brut.

> **Note**
>
> C'est le seul endroit où vous devez entrer des informations d'identification en texte clair pour vous connecter à la base de données de point de terminaison à partir de maintenant.

4. Dans le champ Texte brut :

   - Pour un secret dont vous affectez l'identité à `SecretsManagerSecretId`, entrez la structure JSON suivante.

```
{
  "username": db_username,
  "password": db_user_password,
  "port": db_port_number,
  "host": db_server_name
}
```

> **Note**
>
> Il s'agit de la liste minimale de membres JSON requis pour authentifier la base de données de point de terminaison. Vous pouvez ajouter n'importe quel paramètre de point de terminaison JSON supplémentaire en tant que membre JSON, tout en minuscules. Toutefois, AWS DMS ignore tout membre JSON supplémentaire pour l'authentification du point de terminaison.

Ici, *db_username* est le nom de l'utilisateur accédant à la base de données, *db_user_password* est le mot de passe de l'utilisateur de base de données, *db_port_number* est le numéro de port permettant d'accéder à la base de données et *db_server_name* est le nom du serveur de base de données (adresse) sur le web, comme dans l'exemple suivant.

```
{
  "username": "admin",
  "password": "some_password",
  "port": "8190",
  "host": "oracle101.abcdefghij.us-east-1.rds.amazonaws.com"
```

```
}
```

- Pour un secret dont vous affectez l'identité à `SecretsManagerOracleAsmSecretId`, entrez la structure JSON suivante.

```
{
  "asm_user": asm_username,
  "asm_password": asm_user_password,
  "asm_server": asm_server_name
}
```

> **Note**
>
> Il s'agit de la liste minimale de membres JSON requis pour authentifier Oracle ASM pour un point de terminaison Oracle. Il s'agit également de la liste complète que vous pouvez spécifier en fonction des paramètres de point de terminaison Oracle ASM disponibles.

Ici, *asm_username* est le nom de l'utilisateur accédant à Oracle ASM, *asm_user_password* est le mot de passe de l'utilisateur Oracle ASM et *asm_server_name* est le nom du serveur Oracle ASM (adresse) sur le web, qui comprend le port, comme dans l'exemple suivant.

```
{
  "asm_user": "oracle_asm_user",
  "asm_password": "oracle_asm_password",
  "asm_server": "oracle101.abcdefghij.us-east-1.rds.amazonaws.com:8190/+ASM"
}
```

5. Sélectionnez une clé de AWS KMS chiffrement pour chiffrer le secret. Vous pouvez accepter la clé de chiffrement par défaut créée pour votre service AWS Secrets Manager ou sélectionner une AWS KMS clé que vous créez.
6. Spécifiez un nom pour faire référence à ce secret et une description facultative. Il s'agit du nom convivial que vous utilisez comme valeur pour `SecretsManagerSecretId` ou `SecretsManagerOracleAsmSecretId`.
7. Si vous souhaitez activer la rotation automatique du secret, vous devez sélectionner ou créer une AWS Lambda fonction autorisée à alterner les informations d'identification du secret, comme décrit. Toutefois, avant de configurer la rotation automatique pour utiliser votre fonction Lambda,

assurez-vous que les paramètres de configuration de la fonction ajoutent les quatre caractères suivants à la valeur de la variable d'environnement `EXCLUDE_CHARACTERS`.

```
;.:+{}
```

AWS DMS n'autorise pas ces caractères dans les mots de passe utilisés pour les informations d'identification des terminaux. En configurant votre fonction Lambda de sorte à les exclure, AWS Secrets Manager ne peut pas générer ces caractères dans le cadre de la rotation des valeurs de mot de passe. Une fois que vous avez défini la rotation automatique pour utiliser votre fonction Lambda, fait AWS Secrets Manager immédiatement pivoter le secret pour valider votre configuration secrète.

> **Note**
>
> Selon la configuration de votre moteur de base de données, il est possible que la base de données ne récupère pas les informations d'identification ayant fait l'objet d'une rotation. Dans ce cas, vous devez redémarrer manuellement la tâche pour actualiser les informations d'identification.

8. Passez en revue et conservez votre secret AWS Secrets Manager. Vous pouvez ensuite rechercher chaque secret par son nom convivial AWS Secrets Manager, puis récupérer l'ARN du secret en tant que valeur `SecretsManagerSecretId` ou`SecretsManagerOracleAsmSecretId`, le cas échéant, pour authentifier l'accès à la connexion à la base de données de votre point de terminaison et à Oracle ASM (le cas échéant).

Pour créer la politique d'accès secret et le rôle permettant de définir votre **`SecretsManagerAccessRoleArn`** ou**`SecretsManagerOracleAsmAccessRoleArn`**, ce qui permet d'accéder AWS DMSAWS Secrets Manager à votre secret approprié

1. Connectez-vous à la console AWS Identity and Access Management (IAM) AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Sélectionnez Politiques, puis Créer une politique.
3. Choisissez JSON et entrez la politique suivante pour permettre d'accéder à votre secret et de le déchiffrer.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": secret_arn,
        },
        {
             "Effect": "Allow",
             "Action": [
                        "kms:Decrypt",
                        "kms:DescribeKey"
                      ],
             "Resource": kms_key_arn,
        }
     ]
}
```

Ici, *secret_arn* est l'ARN de votre secret, que vous pouvez obtenir à partir de `SecretsManagerSecretId` ou de `SecretsManagerOracleAsmSecretId` selon le cas, et *kms_key_arn* est l'ARN de la clé AWS KMS que vous utilisez pour chiffrer votre secret, comme dans l'exemple suivant.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "arn:aws:secretsmanager:us-
east-2:123456789012:secret:MySQLTestSecret-qeHamH"
        },
        {
             "Effect": "Allow",
             "Action": [
                        "kms:Decrypt",
                        "kms:DescribeKey"
                      ],
             "Resource": "arn:aws:kms:us-
east-2:123456789012:key/761138dc-0542-4e58-947f-4a3a8458d0fd"
        }
     ]
}
```

> **Note**
>
> Si vous utilisez la clé de chiffrement par défaut créée par AWS Secrets Manager, il n'est pas nécessaire de spécifier les AWS KMS autorisations pour*kms_key_arn*.
> Si vous souhaitez que votre politique donne accès aux deux secrets, il vous suffit de spécifier un objet de ressource JSON supplémentaire pour l'autre *secret_arn*.
> Si votre secret se trouve dans un autre compte, le rôle `SecretsManagerAccessRoleArn` a besoin d'une politique supplémentaire pour vérifier le secret entre comptes. Pour de tels cas d'utilisation, ajoutez l'action `secretsmanager:DescribeSecret` à la politique. Pour plus de détails sur la configuration d'un secret multicompte, consultez la section Permissions to AWS Secrets Manager secrets secrets pour les utilisateurs d'un autre compte.

4. Vérifiez et créez la politique avec un nom convivial et une description facultative.
5. Choisissez Rôles, puis Créer un rôle.
6. Choisissez Service AWS comme type d'entité de confiance.
7. Choisissez DMS dans la liste des services comme service de confiance, puis choisissez Suivant : Autorisations.
8. Recherchez et attachez la politique que vous avez créée à l'étape 4, puis ajoutez des balises et passez en revue votre rôle. À ce stade, modifiez les relations de confiance du rôle afin d'utiliser votre directeur de service AWS DMS régional comme entité de confiance. Ce principal a le format suivant.

```
dms.region-name.amazonaws.com
```

   Ici, *region-name* est le nom de votre région, par exemple `us-east-1`. Voici donc un principe de service AWS DMS régional pour cette région.

```
dms.us-east-1.amazonaws.com
```

9. Après avoir modifié l'entité de confiance associée au rôle, créez le rôle avec un nom convivial et une description facultative. Vous pouvez désormais rechercher votre nouveau rôle par son nom convivial dans IAM, puis récupérer l'ARN du rôle sous forme de valeur `SecretsManagerAccessRoleArn` ou `SecretsManagerOracleAsmAccessRoleArn` pour authentifier la connexion à la base de données de votre point de terminaison.

Pour utiliser Secrets Manager avec une instance de réplication dans un sous-réseau privé

1. Créez un point de terminaison de VPC Secrets Manager et notez le DNS du point de terminaison. Pour plus d'informations sur la création d'un point de terminaison de VPC Secrets Manager, consultez Connexion à Secrets Manager via un point de terminaison de VPC dans le Guide de l'utilisateur AWS Secrets Manager.
2. Attachez le groupe de sécurité de l'instance de réplication au point de terminaison de VPC Secrets Manager.
3. Pour les règles de sortie du groupe de sécurité de l'instance de réplication, autorisez l'ensemble du trafic pour la destination `0.0.0.0/0`.
4. Définissez l'attribut de connexion supplémentaire du point de terminaison `secretsManagerEndpointOverride=`*`secretsManager endpoint DNS`* de sorte à fournir le DNS du point de terminaison de VPC Secrets Manager, comme illustré dans l'exemple suivant.

```
secretsManagerEndpointOverride=vpce-1234a5678b9012c-12345678.secretsmanager.eu-
west-1.vpce.amazonaws.com
```

Utilisation des rôles liés aux services pour AWS DMS

AWS Database Migration Service utilise des rôles AWS Identity and Access Management (IAM) liés à un service. Un rôle lié à un service est un type unique de rôle IAM lié directement à AWS DMS. Les rôles liés à un service sont prédéfinis par AWS DMS et comprennent toutes les autorisations nécessaires au service pour appeler d'autres services AWS en votre nom.

Un rôle lié à un service permet d'utiliser AWS DMS plus facilement, car vous n'avez pas besoin d'ajouter manuellement les autorisations requises. AWS DMS définit les autorisations de ses rôles liés à un service et, sauf définition contraire, seul AWS DMS peut endosser ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Vos ressources AWS DMS sont ainsi protégées, car vous ne pouvez pas involontairement supprimer l'autorisation d'accéder aux ressources.

Pour plus d'informations sur les autres services prenant en charge les rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM et recherchez les services présentant la mention

Yes (Oui) dans la colonne Service-linked roles (Rôles liés à un service). Sélectionnez un Oui ayant un lien pour consulter la documentation du rôle lié à un service, pour ce service.

Rôles liés à un service pour les fonctionnalités AWS DMS

Rubriques

- Rôles liés à un service pour AWS DMS Fleet Advisor
- Rôle lié à un service pour AWS DMS sans serveur

Rôles liés à un service pour AWS DMS Fleet Advisor

AWS DMS Fleet Advisor utilise le rôle lié à un service nommé AWSServiceRoleForDMSFleetAdvisor pour gérer les métriques Amazon CloudWatch. Ce rôle lié à un service est attaché à la politique gérée suivante : `AWSDMSFleetAdvisorServiceRolePolicy`. Pour connaître les mises à jour de cette politique, consultez Politiques AWS gérées pour AWS Database Migration Service.

Le rôle lié à un service AWSServiceRoleForDMSFleetAdvisor approuve les services suivants pour assumer le rôle :

- `dms-fleet-advisor.amazonaws.com`

La politique d'autorisations de rôle nommée AWSDMSFleetAdvisorServiceRolePolicy permet à AWS DMS Fleet Advisor d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `cloudwatch:PutMetricData` sur `all AWS resources`

  Cette autorisation permet aux principaux de publier des points de données de métrique dans Amazon CloudWatch. AWS DMS Fleet Advisor requiert cette autorisation pour afficher des graphiques contenant des métriques de base de données issues de CloudWatch.

L'exemple de code suivant montre la politique AWSDMSFleetAdvisorServiceRolePolicy que vous utilisez pour créer le rôle AWSDMSFleetAdvisorServiceRolePolicy.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Resource": "*",
```

```
        "Action": "cloudwatch:PutMetricData",
        "Condition": {
            "StringEquals": {
                "cloudwatch:namespace": "AWS/DMS/FleetAdvisor"
            }
        }
    }
}
```

Vous devez configurer les autorisations de sorte à permettre à une entité IAM, comme un utilisateur, un groupe ou un rôle, de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez Autorisations de rôles liés à un service dans le Guide de l'utilisateur IAM.

Création d'un rôle lié à un service pour AWS DMS Fleet Advisor

Vous pouvez utiliser la console IAM pour créer un rôle lié à un service avec le cas d'utilisation DMS – Fleet Advisor. Dans l'interface AWS CLI ou l'API AWS, créez un rôle lié à un service avec le nom de service `dms-fleet-advisor.amazonaws.com`. Pour de plus amples informations, consultez Création d'un rôle lié à un service dans le Guide de l'utilisateur IAM. Si vous supprimez ce rôle lié à un service, vous pouvez utiliser ce même processus pour créer le rôle à nouveau.

Assurez-vous de créer ce rôle avant de créer un collecteur de données. DMS Fleet Advisor utilise ce rôle pour afficher des graphiques avec des métriques de base de données dans AWS Management Console. Pour de plus amples informations, veuillez consulter Création d'un collecteur de données.

Modification d'un rôle lié à un service pour AWS DMS Fleet Advisor

AWS DMS ne vous permet pas de modifier le rôle lié à un service AWSServiceRoleForDMSFleetAdvisor. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence au rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour en savoir plus, consultez Modification d'un rôle lié à un service dans le guide de l'utilisateur IAM.

Suppression d'un rôle lié à un service pour AWS DMS Fleet Advisor

Si vous n'avez plus besoin d'utiliser une fonction ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. Ainsi, aucune entité inutilisée n'est surveillée ou gérée activement. Cependant, vous devez nettoyer les ressources de votre rôle lié à un service avant de pouvoir les supprimer manuellement.

> **Note**
>
> Si le service AWS DMS utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression peut échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer les ressources AWS DMS utilisées par AWSServiceRoleForDMSFleetAdvisor

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS DMS à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Dans le volet de navigation, choisissez Collecteurs de données sous Découvrir. La page Collecteurs de données s'ouvre.
3. Choisissez votre collecteur de données, puis choisissez Supprimer.
4. Pour confirmer la suppression, entrez le nom du collecteur de données dans le champ d'entrée du texte. Ensuite, choisissez Supprimer.

> **Important**
>
> Lorsque vous supprimez un collecteur de données DMS, DMS Fleet Advisor supprime toutes les bases de données de l'inventaire que vous avez découvertes à l'aide de ce collecteur.

Après avoir supprimé tous les collecteurs de données, vous pouvez supprimer le rôle lié à un service.

Pour supprimer manuellement le rôle lié à un service à l'aide d'IAM

Utilisez la console IAM, AWS CLI ou l'API AWS pour supprimer le rôle lié à un service AWSServiceRoleForDMSFleetAdvisor. Pour plus d'informations, consultez Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Régions prises en charge pour les rôles liés à un service AWS DMS Fleet Advisor

AWS DMS Fleet Advisor prend en charge l'utilisation des rôles liés à un service dans toutes les régions où le service est disponible. Pour de plus amples informations, veuillez consulter Régions AWS prises en charge.

Rôle lié à un service pour AWS DMS sans serveur

AWS DMS Serverless utilise le rôle lié au service nommé. AWSServiceRoleForDMSServerless AWS DMS utilise ce rôle lié au service pour créer et gérer des AWS DMS ressources en votre nom, telles que les métriques Amazon CloudWatch . AWS DMS utilise ce rôle afin que vous n'ayez à vous préoccuper que des réplications. Ce rôle lié à un service est attaché à la politique gérée suivante : `AWSDMSServerlessServiceRolePolicy`. Pour connaître les mises à jour de cette politique, consultez Politiques AWS gérées pour AWS Database Migration Service.

Le rôle AWSServiceRoleForDMSServerless lié à un service fait confiance aux services suivants pour assumer le rôle :

- `dms.amazonaws.com`

L'exemple de code suivant montre la AWSDMSServerlessServiceRolePolicy politique que vous utilisez pour créer le AWSServiceRoleForDMSServerless rôle.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "id0",
            "Effect": "Allow",
            "Action": [
                "dms:CreateReplicationInstance",
                "dms:CreateReplicationTask"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "dms:req-tag/ResourceCreatedBy": "DMSServerless"
                }
            }
        },
        {
            "Sid": "id1",
            "Effect": "Allow",
            "Action": [
                "dms:DescribeReplicationInstances",
                "dms:DescribeReplicationTasks"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "id2",
            "Effect": "Allow",
            "Action": [
                "dms:StartReplicationTask",
                "dms:StopReplicationTask",
                "dms:DeleteReplicationTask",
                "dms:DeleteReplicationInstance"
            ],
            "Resource": [
                "arn:aws:dms:*:*:rep:*",
                "arn:aws:dms:*:*:task:*"
            ],
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "aws:ResourceTag/ResourceCreatedBy": "DMSServerless"
                }
            }
        },
        {
            "Sid": "id3",
            "Effect": "Allow",
            "Action": [
                "dms:TestConnection",
                "dms:DeleteConnection"
            ],
            "Resource": [
                "arn:aws:dms:*:*:rep:*",
                "arn:aws:dms:*:*:endpoint:*"
            ]
        }
    ]
}
```

Vous devez configurer les autorisations de sorte à permettre à une entité IAM, comme un utilisateur, un groupe ou un rôle, de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez Autorisations de rôles liés à un service dans le Guide de l'utilisateur IAM.

Création d'un rôle lié à un service pour AWS DMS sans serveur

Lorsque vous créez une réplication, AWS DMS Serverless crée par programmation un rôle lié à un service AWS DMS sans serveur. Vous pouvez afficher ce rôle dans la console IAM. Vous pouvez également choisir de créer ce rôle manuellement. Pour créer le rôle manuellement, utilisez la console IAM pour créer un rôle lié à un service avec le cas d'utilisation du DMS. Dans l'API AWS CLI ou dans l' AWS API, créez un rôle lié à un service en utilisant le `dms.amazonaws.com` nom du service. Pour plus d'informations, consultez Création d'un rôle lié à un service dans le Guide de l'utilisateur IAM. Si vous supprimez ce rôle lié à un service, vous pouvez utiliser ce même processus pour créer le rôle à nouveau.

> **Note**
>
> Si vous supprimez un rôle alors que vous avez des réplications dans votre compte, la réplication échoue.

Modification d'un rôle lié à un service pour AWS DMS sans serveur

AWS DMS ne vous permet pas de modifier le rôle AWSServiceRoleForDMSServerless lié au service. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence au rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour en savoir plus, consultez Modification d'un rôle lié à un service dans le guide de l'utilisateur IAM.

Suppression d'un rôle lié à un service pour AWS DMS sans serveur

Si vous n'avez plus besoin d'utiliser une fonction ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. Ainsi, aucune entité inutilisée n'est surveillée ou gérée activement. Cependant, vous devez nettoyer les ressources de votre rôle lié à un service avant de pouvoir les supprimer manuellement.

> **Note**
>
> Si le AWS DMS service utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression risque d'échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer AWS DMS les ressources utilisées par AWSServiceRoleForDMSServerless

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Dans le volet de navigation, choisissez Sans serveur sous Découvrir. La page Sans serveur s'ouvre.
3. Choisissez votre réplication sans serveur, puis choisissez Supprimer.
4. Pour confirmer la suppression, entrez le nom de la réplication sans serveur dans le champ d'entrée du texte. Ensuite, choisissez Supprimer.

Après avoir supprimé toutes les réplications sans serveur, vous pouvez supprimer le rôle lié à un service.

Pour supprimer manuellement le rôle lié à un service à l'aide d'IAM

Utilisez la console IAM, le AWS CLI, ou l' AWS API pour supprimer le rôle lié au AWSServiceRoleForDMSServerless service. Pour plus d'informations, consultez Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Régions prises en charge pour les rôles liés à un service AWS DMS sans serveur

AWS DMS Serverless prend en charge l'utilisation de rôles liés au service dans toutes les régions où le service est disponible.

Résolution des problèmes AWS Database Migration Service d'identité et d'accès

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec AWS DMS IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans AWS DMS
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je suis administrateur et je souhaite autoriser d'autres personnes à accéder AWS DMS
- Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes AWS DMS ressources

Je ne suis pas autorisé à effectuer une action dans AWS DMS

S'il vous AWS Management Console indique que vous n'êtes pas autorisé à effectuer une action, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni votre nom d'utilisateur et votre mot de passe.

L'exemple d'erreur suivant se produit lorsque l'utilisateur `mateojackson` IAM essaie d'utiliser la console pour afficher les détails d'un point de terminaison AWS DMS mais ne dispose `dms:DescribeEndpoint` pas des autorisations nécessaires.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 dms:DescribeEndpoint on resource: my-postgresql-target
```

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses stratégies pour lui permettre d'accéder à la ressource de point de terminaison `my-postgresql-target` à l'aide de l'action `dms:DescribeEndpoint`.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter `iam:PassRole` l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à AWS DMS.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour exécuter une action dans AWS DMS. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les stratégies de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je suis administrateur et je souhaite autoriser d'autres personnes à accéder AWS DMS

Pour autoriser d'autres personnes à y accéder AWS DMS, vous devez créer une entité IAM (utilisateur ou rôle) pour la personne ou l'application qui a besoin d'un accès. Ils utiliseront les informations d'identification de cette entité pour accéder à AWS. Vous devez ensuite associer une politique à l'entité qui leur accorde les autorisations appropriées dans AWS DMS.

Pour démarrer immédiatement, veuillez consulter Création de votre premier groupe et utilisateur délégué IAM dans le Guide de l'utilisateur IAM.

Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes AWS DMS ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises AWS DMS en charge, consultezComment AWS Database Migration Service fonctionne avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès intercompte, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

Autorisations IAM nécessaires pour utiliser AWS DMS

Vous devez utiliser certaines autorisations et certains rôles IAM pour pouvoir utiliser AWS DMS. Si vous êtes connecté en tant qu'utilisateur IAM et que vous souhaitez l'utiliser AWS DMS, l'administrateur de votre compte doit associer la politique décrite dans cette section à l'utilisateur, au groupe ou au rôle IAM que vous utilisez pour exécuter. AWS DMS Pour plus d'informations sur les autorisations IAM, consultez le Guide de l'utilisateur IAM.

La politique suivante vous donne accès à d'autres services Amazon tels que IAM AWS DMS, Amazon EC2 et Amazon AWS KMS, ainsi que des autorisations pour certaines actions nécessaires. CloudWatch CloudWatchsurveille votre AWS DMS migration en temps réel et collecte et suit les indicateurs qui indiquent la progression de votre migration. Vous pouvez utiliser CloudWatch les journaux pour résoudre les problèmes liés à une tâche.

> **Note**
>
> Vous pouvez restreindre davantage l'accès aux AWS DMS ressources à l'aide du balisage. Pour plus d'informations sur la restriction de l'accès aux AWS DMS ressources à l'aide du balisage, consultezContrôle précis des accès à l'aide des noms de ressources et des balises.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "dms:*",
            "Resource": "arn:aws:dms:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "iam:GetRole",
                "iam:PassRole",
                "iam:CreateRole",
                "iam:AttachRolePolicy"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:Get*",
                "cloudwatch:List*"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:FilterLogEvents",
                "logs:GetLogEvents"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
    ]
}
```

L'analyse des autorisations suivantes peut vous aider à mieux comprendre pourquoi chacune d'elles est nécessaire.

La section suivante est requise pour permettre à l'utilisateur d'appeler des opérations AWS DMS d'API.

```
{
            "Effect": "Allow",
            "Action": "dms:*",
            "Resource": "arn:aws:dms:region:account:resourcetype/id"
}
```

La section suivante est requise pour permettre à l'utilisateur de répertorier ses AWS KMS clés et alias disponibles à afficher dans la console. Cette entrée n'est pas obligatoire si vous connaissez le nom de ressource Amazon (ARN) de la clé KMS et que vous utilisez uniquement le AWS Command Line Interface (AWS CLI).

```
{
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

La section suivante est requise lorsque certains types de points de terminaison nécessitent la transmission de l'ARN d'un rôle IAM avec le point de terminaison. En outre, si les AWS DMS rôles requis ne sont pas créés à l'avance, la AWS DMS console peut créer le rôle. Si tous les rôles sont configurés à l'avance, seuls `iam:GetRole` et `iam:PassRole` sont nécessaires. Pour plus d'informations sur les rôles , consultez Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS.

```
{
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole",
                "iam:CreateRole",
                "iam:AttachRolePolicy"
```

```
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

La section suivante est obligatoire car elle AWS DMS doit créer l'instance Amazon EC2 et configurer le réseau pour l'instance de réplication créée. Comme ces ressources existent dans le compte du client, il est obligatoire de pouvoir effectuer ces actions au nom du client.

```
{
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

La section suivante est obligatoire pour permettre à l'utilisateur d'afficher les métriques de l'instance de réplication.

```
{
            "Effect": "Allow",
            "Action": [
                "cloudwatch:Get*",
                "cloudwatch:List*"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

Cette section est obligatoire pour permettre à l'utilisateur d'afficher les journaux de réplication.

```
{
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
```

```
                "logs:DescribeLogStreams",
                "logs:FilterLogEvents",
                "logs:GetLogEvents"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

La AWS DMS console crée plusieurs rôles qui sont automatiquement associés à votre AWS compte lorsque vous utilisez la AWS DMS console. Si vous utilisez le AWS Command Line Interface (AWS CLI) ou l' AWS DMS API pour votre migration, vous devez ajouter ces rôles à votre compte. Pour plus d'informations sur l'ajout de ces rôles, consultez Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS.

Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS

Si vous utilisez l'API AWS CLI ou l' AWS DMS API pour la migration de votre base de données, vous devez ajouter trois rôles IAM à votre AWS compte avant de pouvoir utiliser les fonctionnalités de AWS DMS. Deux d'entre eux sont `dms-vpc-role` et `dms-cloudwatch-logs-role`. Si vous utilisez Amazon Redshift comme base de données cible, vous devez également ajouter le rôle IAM `dms-access-for-endpoint` à votre compte. AWS

Les mises à jour des stratégies gérées sont automatiques. Si vous utilisez une stratégie personnalisée avec les rôles IAM, veillez à vérifier régulièrement la disponibilité des mises à jour de la stratégie gérée dans cette documentation. Vous pouvez afficher les détails de la stratégie gérée grâce à une combinaison des commandes `get-policy` et `get-policy-version`.

Par exemple, la commande `get-policy` suivante récupère les informations relatives au rôle IAM spécifié.

```
aws iam get-policy --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonDMSVPCManagementRole
```

Les informations renvoyées par la commande sont les suivantes.

```
{
    "Policy": {
        "PolicyName": "AmazonDMSVPCManagementRole",
        "Description": "Provides access to manage VPC settings for AWS managed customer
 configurations",
        "CreateDate": "2015-11-18T16:33:19Z",
        "AttachmentCount": 1,
        "IsAttachable": true,
        "PolicyId": "ANPAJHKIGMBQI4AEFFSYO",
        "DefaultVersionId": "v3",
        "Path": "/service-role/",
        "Arn": "arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole",
        "UpdateDate": "2016-05-23T16:29:57Z"
    }
}
```

La commande `get-policy-version` suivante récupère les informations relatives à la stratégie IAM.

```
aws iam get-policy-version --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonDMSVPCManagementRole --version-id v3
```

Les informations renvoyées par la commande sont les suivantes.

```
{
    "PolicyVersion": {
        "CreateDate": "2016-05-23T16:29:57Z",
        "VersionId": "v3",
        "Document": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "ec2:CreateNetworkInterface",
                    "ec2:DescribeAvailabilityZones",
                    "ec2:DescribeInternetGateways",
                    "ec2:DescribeSecurityGroups",
                    "ec2:DescribeSubnets",
                    "ec2:DescribeVpcs",
                    "ec2:DeleteNetworkInterface",
                    "ec2:ModifyNetworkInterfaceAttribute"
                ],
                "Resource": "arn:aws:service:region:account:resourcetype/id",
                "Effect": "Allow"
            }
        ]
    },
    "IsDefaultVersion": true
    }
}
```

Vous pouvez utiliser les mêmes commandes pour obtenir des informations sur la stratégie gérée `AmazonDMSCloudWatchLogsRole` et `AmazonDMSRedshiftS3Role`.

> **Note**
>
> Si vous utilisez la AWS DMS console pour la migration de votre base de données, ces rôles sont automatiquement ajoutés à votre AWS compte.

Les procédures suivantes créent les rôles IAM `dms-vpc-role`, `dms-cloudwatch-logs-role` et `dms-access-for-endpoint`.

Pour créer le rôle dms-vpc-role IAM à utiliser avec l'API AWS CLI or AWS DMS

1. Créez un fichier JSON contenant la politique IAM suivante. Nommez le fichier JSON `dmsAssumeRolePolicyDocument.json`.

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
     "Effect": "Allow",
     "Principal": {
        "Service": "dms.amazonaws.com"
     },
   "Action": "sts:AssumeRole"
   }
 ]
}
```

   Créez le rôle à l' AWS CLI aide de la commande suivante.

```
aws iam create-role --role-name dms-vpc-role --assume-role-policy-document file://
dmsAssumeRolePolicyDocument.json
```

2. Attachez la stratégie `AmazonDMSVPCManagementRole` à `dms-vpc-role` en utilisant la commande suivante.

```
aws iam attach-role-policy --role-name dms-vpc-role --policy-arn
 arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole
```

Pour créer le rôle dms-cloudwatch-logs-role IAM à utiliser avec l'API AWS CLI or AWS DMS

1. Créez un fichier JSON contenant la politique IAM suivante. Nommez le fichier JSON `dmsAssumeRolePolicyDocument2.json`.

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
     "Effect": "Allow",
     "Principal": {
        "Service": "dms.amazonaws.com"
     },
   "Action": "sts:AssumeRole"
   }
 ]
}
```

   Créez le rôle à l' AWS CLI aide de la commande suivante.

```
aws iam create-role --role-name dms-cloudwatch-logs-role --assume-role-policy-
document file://dmsAssumeRolePolicyDocument2.json
```

2. Attachez la stratégie `AmazonDMSCloudWatchLogsRole` à `dms-cloudwatch-logs-role` en utilisant la commande suivante.

```
aws iam attach-role-policy --role-name dms-cloudwatch-logs-role --policy-arn
 arn:aws:iam::aws:policy/service-role/AmazonDMSCloudWatchLogsRole
```

Si vous utilisez Amazon Redshift en tant que base de données cible, vous devez créer le rôle IAM `dms-access-for-endpoint` pour fournir l'accès à Amazon S3.

Pour créer le rôle dms-access-for-endpoint IAM à utiliser avec Amazon Redshift en tant que base de données cible

1. Créez un fichier JSON contenant la politique IAM suivante. Nommez le fichier JSON `dmsAssumeRolePolicyDocument3.json`.

```
 {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "1",
      "Effect": "Allow",
      "Principal": {
        "Service": "dms.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Sid": "2",
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Créez le rôle à l' AWS CLI aide de la commande suivante.

```
  aws iam create-role --role-name dms-access-for-endpoint --assume-role-policy-
document file://dmsAssumeRolePolicyDocument3.json
```

3. Attachez la stratégie `AmazonDMSRedshiftS3Role` au rôle `dms-access-for-endpoint` en utilisant la commande suivante.

```
aws iam attach-role-policy --role-name dms-access-for-endpoint \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonDMSRedshiftS3Role
```

Vous devriez maintenant avoir mis en place les politiques IAM pour utiliser l' AWS DMS API AWS CLI or.

Prévention du problème de l'adjoint confus entre services

Le problème de l'adjoint confus est un problème de sécurité dans lequel une entité qui n'a pas l'autorisation d'effectuer une action peut contraindre une entité plus privilégiée à effectuer cette action. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé d'accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition `aws:SourceAccount`globale `aws:SourceArn`et les clés contextuelles dans les politiques de ressources afin de limiter les autorisations qui AWS Database Migration Service accordent un autre service à la ressource. Si la valeur `aws:SourceArn` ne contient pas l'ID de compte, tel qu'un nom d'instance de réplication AWS DMS (ARN), vous devez utiliser les deux clés de contexte de condition globale pour limiter les autorisations. Si vous utilisez les deux clés de contexte de condition globale et que la valeur `aws:SourceArn` contient l'ID de compte, la valeur `aws:SourceAccount` et le compte dans la valeur `aws:SourceArn` doivent utiliser le même ID de compte lorsqu'ils sont utilisés dans la même instruction de politique. Utilisez `aws:SourceArn` si vous souhaitez qu'une seule ressource soit associée à l'accès entre services. Utilisez `aws:SourceAccount` si vous souhaitez autoriser l'association d'une ressource de ce compte à l'utilisation interservices.

AWS DMS prend en charge les options secondaires confuses à partir de la version 3.4.7 et supérieure. Pour plus d'informations, consultez AWS Notes de mise à jour du Database Migration Service 3.4.7. Si votre instance de réplication utilise AWS DMS version 3.4.6 ou antérieure, assurez-vous de passer à la dernière version avant de définir les options d'adjoint confus.

Le moyen le plus efficace de se protéger du problème de l'adjoint désorienté consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (`*`) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:dms:*:123456789012:rep:*`.

Rubriques

- Rôles IAM à utiliser avec l' AWS DMS API pour prévenir la confusion entre les services
- Politique IAM visant à stocker les évaluations de contrôle en amont dans Amazon S3 pour la prévention du problème de l'adjoint confus entre services

- Utilisation d'Amazon DynamoDB comme point de terminaison cible pour la prévention de la confusion AWS DMS entre les services

Rôles IAM à utiliser avec l' AWS DMS API pour prévenir la confusion entre les services

Pour utiliser l'API AWS CLI ou l' AWS DMS API pour la migration de votre base de données, vous devez ajouter les rôles `dms-vpc-role` et `dms-cloudwatch-logs-role` IAM à votre AWS compte avant de pouvoir utiliser les fonctionnalités de AWS DMS. Pour plus d'informations, consultez Création des rôles IAM à utiliser avec l'API AWS CLI and AWS DMS.

L'exemple suivant illustre les politiques d'utilisation du rôle `dms-vpc-role` avec l'instance de réplication `my-replication-instance`. Utilisez ces politiques pour prévenir le problème de l'adjoint confus.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "dms.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "your_account_id"
                },
                "ArnEqual": {
                    "AWS:SourceArn": "arn:aws:dms:your_region:your_account_id:rep:my-
replication-instance"
                }
            }
        }
    ]
}
```

Politique IAM visant à stocker les évaluations de contrôle en amont dans Amazon S3 pour la prévention du problème de l'adjoint confus entre services

Pour stocker les résultats de la pré-évaluation dans votre compartiment S3, vous créez une politique IAM qui permet à AWS DMS de gérer les objets dans Amazon S3. Pour plus d'informations, consultez Création de ressources IAM .

L'exemple suivant montre une politique de confiance comportant des conditions d'adjoint confuses définies pour un rôle IAM qui permet d'accéder AWS DMS à toutes les tâches et à toutes les évaluations sous un compte utilisateur spécifié.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "dms.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "your_account_id"
                },
                "ArnLike": {
                    "AWS:SourceArn": [
                        "arn:aws:dms:your_region:your_account_id:assessment-run:*",
                        "arn:aws:dms:region:your_account_id:task:*"
                    ]
                }
            }
        }
    ]
}
```

Utilisation d'Amazon DynamoDB comme point de terminaison cible pour la prévention de la confusion AWS DMS entre les services

Pour utiliser Amazon DynamoDB comme point de terminaison cible pour la migration de votre base de données, vous devez créer le rôle IAM qui AWS DMS permet d'assumer et d'accorder l'accès aux

tables DynamoDB. Utilisez ensuite ce rôle lorsque vous créez votre point de terminaison DynamoDB cible dans AWS DMS. Pour plus d'informations, consultez Utilisation d'Amazon DynamoDB en tant que cible.

L'exemple suivant montre une politique de confiance comportant des conditions secondaires confuses définies sur un rôle IAM qui permet à tous les AWS DMS points de terminaison d'accéder aux tables DynamoDB.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "dms.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "your_account_id"
                },
                "ArnLike": {
                    "AWS:SourceArn":
 "arn:aws:dms:your_region:your_account_id:endpoint:*"
                }
            }
        }
    ]
}
```

Politiques AWS gérées pour AWS Database Migration Service

Rubriques

- Politique gérée AWS : AWSDMSServerlessServiceRolePolicy
- Politique gérée AWS : AmazonDMSCloudWatchLogsRole
- Politique gérée AWS : AWSDMSFleetAdvisorServiceRolePolicy
- Mises à jour AWS DMS vers des politiques gérées par AWS

Politique gérée AWS : AWSDMSServerlessServiceRolePolicy

Cette politique est attachée au rôle `AWSServiceRoleForDMSServerless`, qui permet à AWS DMS d'effectuer des actions en votre nom. Pour de plus amples informations, veuillez consulter Rôle lié à un service pour AWS DMS sans serveur.

Cette politique accorde au contributeur des autorisations qui permettent à AWS DMS de gérer les ressources de réplication.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- `dms` : permet aux principaux d'interagir avec les ressources AWS DMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "id0",
            "Effect": "Allow",
            "Action": [
                "dms:CreateReplicationInstance",
                "dms:CreateReplicationTask"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "dms:req-tag/ResourceCreatedBy": "DMSServerless"
                }
            }
        },
        {
            "Sid": "id1",
            "Effect": "Allow",
            "Action": [
                "dms:DescribeReplicationInstances",
                "dms:DescribeReplicationTasks"
            ],
            "Resource": "*"
        },
        {
```

```
            "Sid": "id2",
            "Effect": "Allow",
            "Action": [
                "dms:StartReplicationTask",
                "dms:StopReplicationTask",
                "dms:DeleteReplicationTask",
                "dms:DeleteReplicationInstance"
            ],
            "Resource": [
                "arn:aws:dms:*:*:rep:*",
                "arn:aws:dms:*:*:task:*"
            ],
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "aws:ResourceTag/ResourceCreatedBy": "DMSServerless"
                }
            }
        },
        {
            "Sid": "id3",
            "Effect": "Allow",
            "Action": [
                "dms:TestConnection",
                "dms:DeleteConnection"
            ],
            "Resource": [
                "arn:aws:dms:*:*:rep:*",
                "arn:aws:dms:*:*:endpoint:*"
            ]
        }
    ]
}
```

Politique gérée AWS : AmazonDMSCloudWatchLogsRole

Cette politique est attachée au rôle `dms-cloudwatch-logs-role`, qui permet à AWS DMS d'effectuer des actions en votre nom. Pour de plus amples informations, veuillez consulter Utilisation des rôles liés aux services pour AWS DMS.

Cette politique accorde au contributeur des autorisations qui permettent à AWS DMS de publier les journaux de réplication sur CloudWatch Logs.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- `logs` : permet aux principaux de publier des journaux dans CloudWatch Logs. Cette autorisation est requise pour qu'AWS DMS puisse utiliser CloudWatch pour afficher les journaux de réplication.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDescribeOnAllLogGroups",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "AllowDescribeOfAllLogStreamsOnDmsTasksLogGroup",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:dms-tasks-*",
                "arn:aws:logs:*:*:log-group:dms-serverless-replication-*"
            ]
        },
        {
            "Sid": "AllowCreationOfDmsLogGroups",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:dms-tasks-*",
                "arn:aws:logs:*:*:log-group:dms-serverless-replication-*:log-stream:"
            ]
        },
```

```
        {
            "Sid": "AllowCreationOfDmsLogStream",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:dms-tasks-*:log-stream:dms-task-*",
                "arn:aws:logs:*:*:log-group:dms-serverless-replication-*:log-
stream:dms-serverless-*"
            ]
        },
        {
            "Sid": "AllowUploadOfLogEventsToDmsLogStream",
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:dms-tasks-*:log-stream:dms-task-*",
                "arn:aws:logs:*:*:log-group:dms-serverless-replication-*:log-
stream:dms-serverless-*"
            ]
        }
    ]
}
```

Politique gérée AWS : AWSDMSFleetAdvisorServiceRolePolicy

Vous ne pouvez pas attacher AWSDMSFleetAdvisorServiceRolePolicy à vos entités IAM. Cette politique est attachée à un rôle lié au service qui permet à AWS DMS Fleet Advisor d'effectuer des actions en votre nom. Pour de plus amples informations, veuillez consulter Utilisation des rôles liés aux services pour AWS DMS.

Cette politique accorde au contributeur des autorisations qui permettent à AWS DMS Fleet Advisor de publier des métriques Amazon CloudWatch.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- `cloudwatch` : permet aux principaux de publier des points de données de métrique dans Amazon CloudWatch. Cette autorisation est requise pour qu'AWS DMS Fleet Advisor puisse utiliser CloudWatch pour afficher des graphiques contenant des métriques de base de données.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Resource": "*",
        "Action": "cloudwatch:PutMetricData",
        "Condition": {
            "StringEquals": {
                "cloudwatch:namespace": "AWS/DMS/FleetAdvisor"
            }
        }
    }
}
```

Mises à jour AWS DMS vers des politiques gérées par AWS

Consultez le détail des mises à jour des politiques gérées par AWS pour AWS DMS depuis que ce service a commencé à suivre ces modifications. Pour obtenir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS sur la page d'historique du document AWS DMS.

Modification	Description	Date
AWSDMSServerlessServiceRolePolicy : nouvelle politique	AWS DMS a ajouté le rôle `AWSDMSServerlessServiceRolePolicy` pour permettre à AWS DMS de créer et de gérer des services en votre nom, tels que la publication de métriques Amazon CloudWatch.	22 mai 2023

Modification	Description	Date
AmazonDMSCloudWatchLogsRole : modification	AWS DMS a ajouté l'ARN pour les ressources sans serveur à chacune des autorisations accordées, afin de permettre le chargement des journaux de réplication AWS DMS à partir des configurations de réplication sans serveur vers CloudWatch Logs.	22 mai 2023
AWSDMSFleetAdvisorServiceRolePolicy : nouvelle politique	AWS DMS Fleet Advisor a ajouté une nouvelle politique pour permettre la publication de points de données de métriques sur Amazon CloudWatch.	06 mars 2023
AWS DMS a démarré le suivi des modifications	AWS DMS a commencé à suivre les modifications pour ses politiques gérées par AWS.	06 mars 2023

Validation de la conformité pour AWS Database Migration Service

Les auditeurs tiers évaluent la sécurité et la conformité de AWS Database Migration Service dans le cadre de plusieurs programmes de conformité AWS. Ces programmes sont les suivants :

- SOC
- PCI
- ISO
- FedRAMP
- DoD CC SRG
- HIPAA BAA
- MTCS
- CS
- K-ISMS
- ENS High
- OSPAR
- HITRUST CSF

Pour obtenir la liste des services AWS relevant de programmes de conformité spécifiques, consultez Services AWS relevant de programmes de conformité . Pour obtenir des renseignements généraux, consultez Programmes de conformitéAWS .

Vous pouvez télécharger les rapports de l'audit externe avec AWS Artifact. Pour plus d'informations, consultez Téléchargement de rapports dans AWS Artifact.

Votre responsabilité de conformité lors de l'utilisation de AWS DMS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise, ainsi que par la législation et la réglementation applicables. AWS fournit les ressources suivantes pour faciliter le respect de la conformité :

- Guides de démarrage rapide de la sécurité et de la conformité : ces guides de déploiement traitent de considérations architecturales et indiquent comment déployer des environnements de référence axés sur la sécurité et la conformité sur AWS.

- Livre blanc Architecture pour la sécurité et la conformité HIPAA sur Amazon Web Services (langue française non garantie) : ce livre blanc décrit comment les entreprises peuvent utiliser AWS pour créer des applications conformes à la loi HIPAA.
- Ressources de conformité AWS : cet ensemble de manuels et de guides peut s'appliquer à votre secteur et à votre emplacement.
- AWS Config : ce service AWS permet d'évaluer la conformité des configurations de vos ressources par rapport à des pratiques internes, règlementations et autres directives sectorielles.
- AWS Security Hub : ce service AWS fournit une vue complète de votre état de sécurité au sein d'AWS qui vous permet de vérifier votre conformité aux normes du secteur et aux bonnes pratiques de sécurité.

Résilience dans AWS Database Migration Service

L'infrastructure mondiale d'AWS repose sur les Régions AWS et les zones de disponibilité AWS. Les Régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur les régions AWS et les zones de disponibilité, veuillez consulter Infrastructure mondiale AWS.

Outre l'infrastructure AWS globale, AWS DMS fournit une haute disponibilité et une prise en charge du basculement pour une instance de réplication à l'aide d'un déploiement Multi-AZ lorsque vous sélectionnez l'option Multi-AZ.

Dans un déploiement multi-AZ, AWS DMS met en service et gère automatiquement un réplica de secours de l'instance de réplication dans une zone de disponibilité différente. L'instance de réplication principale est répliquée de manière synchrone vers le réplica de secours. Si l'instance de réplication principale échoue ou ne répond plus, l'instance de secours reprend toutes les tâches en cours avec une interruption minimale. Comme l'instance principale réplique constamment son état vers l'instance de secours, un déploiement multi-AZ occasionne une certaine dégradation des performances.

Pour plus d'informations sur le fonctionnement des déploiements Multi-AZ, consultez Utilisation d'une instance AWS DMS de réplication.

Sécurité de l'infrastructure dans AWS Database Migration Service

En tant que service géré, AWS Database Migration Service est protégé par les procédures de sécurité du réseau mondial AWS. Pour plus d'informations sur les services de sécurité AWS et la manière dont AWS protège l'infrastructure, consultez la section Sécurité du cloud AWS. Pour concevoir votre environnement AWS en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le Security Pillar AWS Well-Architected Framework (Pilier de sécurité de l'infrastructure Well-Architected Framework).

Vous utilisez les appels d'API publiés AWS pour accéder à AWS DMS via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et nous recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Vous pouvez appeler ces opérations d'API à partir de n'importe quel emplacement réseau. AWS DMS prend également en charge les stratégies d'accès basées sur les ressources, qui peuvent spécifier des restrictions sur les actions et les ressources, par exemple, en fonction de l'adresse IP source. De plus, vous pouvez utiliser des politiques AWS DMS pour contrôler l'accès à partir de points de terminaison Amazon VPC spécifiques ou de clouds privés virtuels (VPC) spécifiques. En effet, cela permet d'isoler l'accès réseau vers une ressource AWS DMS donnée depuis le VPC spécifique uniquement au sein du réseau AWS. Pour plus d'informations sur l'utilisation de stratégies d'accès basées sur les ressources avec AWS DMS, avec des exemples à l'appui, consultez Contrôle précis des accès à l'aide des noms de ressources et des balises.

Pour limiter vos communications avec AWS DMS au sein d'un seul VPC, vous pouvez créer un point de terminaison d'interface VPC qui vous permettra de vous connecter à AWS DMS via AWS PrivateLink. AWS PrivateLink permet de garantir que tout appel à AWS DMS et ses résultats associés resteront confinés dans le VPC spécifique pour lequel votre point de terminaison d'interface est créé. Vous pouvez ensuite spécifier l'URL de ce point de terminaison d'interface en tant qu'option

pour chaque commande AWS DMS que vous exécutez à l'aide d'AWS CLI ou d'un kit SDK. Cela permet de garantir que l'ensemble de vos communications avec AWS DMS resteront confinées dans le VPC et seront autrement invisibles sur le réseau Internet public.

Pour créer un point de terminaison d'interface afin d'accéder à DMS dans un seul VPC

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. Dans le volet de navigation, choisissez Points de terminaison. Cela ouvre la page Créer des points de terminaison, où vous pouvez créer le point de terminaison d'interface d'un VPC à AWS DMS.
3. Choisissez Services AWS, puis recherchez et choisissez une valeur pour Nom du service, dans ce cas AWS DMS sous la forme suivante.

   ```
   com.amazonaws.region.dms
   ```

   Ici, *region* spécifie la région AWS dans laquelle AWS DMS s'exécute, par exemple `com.amazonaws.us-west-2.dms`.
4. Pour VPC, choisissez le VPC à partir duquel créer le point de terminaison d'interface, par exemple `vpc-12abcd34`.
5. Choisissez une valeur pour Zone de disponibilité et pour ID de sous-réseau. Ces valeurs doivent indiquer un emplacement où le point de terminaison AWS DMS que vous avez choisi peut s'exécuter, par exemple `us-west-2a (usw2-az1)` et `subnet-ab123cd4`.
6. Choisissez Activer le nom DNS pour créer le point de terminaison avec un nom DNS. Ce nom DNS se compose de l'identifiant du point de terminaison (`vpce-12abcd34efg567hij`) accompagné d'un trait d'union et d'une chaîne aléatoire (`ab12dc34`). Ils sont séparés du nom du service par un point dans l'ordre inverse séparé par des points, avec l'ajout de `vpce` (`dms.us-west-2.vpce.amazonaws.com`).

   Par exemple : `vpce-12abcd34efg567hij-ab12dc34.dms.us-west-2.vpce.amazonaws.com`.
7. Pour Groupe de sécurité, choisissez un groupe à utiliser pour le point de terminaison.

   Lorsque vous configurez votre groupe de sécurité, assurez-vous d'autoriser les appels HTTPS sortants en provenance de celui-ci. Pour plus d'informations, consultez Création de groupes de sécurité dans le Guide de l'utilisateur Amazon VPC.

8. Choisissez Accès complet ou une valeur personnalisée pour Politique. Par exemple, vous pouvez choisir une politique personnalisée similaire à la suivante qui restreint l'accès du point de terminaison à certaines actions et ressources.

```
{
  "Statement": [
    {
      "Action": "dms:*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": [
        "dms:ModifyReplicationInstance",
        "dms:DeleteReplicationInstance"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:dms:us-west-2:<account-id>:rep:<replication-instance-
id>",
      "Principal": "*"
    }
  ]
}
```

   Ici, l'exemple de politique autorise n'importe quel appel d'API AWS DMS, à l'exception de la suppression ou de la modification d'une instance de réplication spécifique.

Vous pouvez désormais spécifier une URL formée à l'aide du nom DNS créé à l'étape 6 en option. Vous la spécifiez pour chaque commande CLI AWS DMS ou opération d'API afin d'accéder à l'instance du service à l'aide du point de terminaison d'interface créé. Par exemple, vous pouvez exécuter la commande CLI DMS `DescribeEndpoints` dans ce VPC comme indiqué ci-dessous.

```
$ aws dms describe-endpoints --endpoint-url https://vpce-12abcd34efg567hij-
ab12dc34.dms.us-west-2.vpce.amazonaws.com
```

Si vous activez l'option de DNS privé, vous n'avez pas à spécifier l'URL du point de terminaison dans la demande.

Pour plus d'informations sur la création et l'utilisation des points de terminaison d'interface de VPC (y compris l'activation de l'option de DNS privé), consultez Points de terminaison de VPC d'interface (AWS PrivateLink) dans le Guide de l'utilisateur Amazon VPC.

Contrôle précis des accès à l'aide des noms de ressources et des balises

Vous pouvez utiliser les noms de ressources basés sur l'Amazon Resource Name (ARN) et les balises des ressources pour gérer l'accès aux ressources AWS DMS. Pour ce faire, définissez l'action autorisée ou incluez les instructions conditionnelles dans les stratégies IAM.

Utilisation des noms de ressources pour contrôler l'accès

Vous pouvez créer un compte utilisateur IAM et attribuer une stratégie basée sur l'ARN de la ressource AWS DMS.

La stratégie suivante refuse l'accès à l'instance de réplication AWS DMS ayant l'ARN arn:aws:dms:us-east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dms:*"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:dms:us-east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV"
        }
    ]
}
```

Par exemple, les commandes suivantes échouent lorsque la stratégie est en vigueur.

```
$ aws dms delete-replication-instance
   --replication-instance-arn "arn:aws:dms:us-
east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV"

A client error (AccessDeniedException) occurred when calling the
 DeleteReplicationInstance
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
```

```
dms:DeleteReplicationInstance on resource: arn:aws:dms:us-
east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV

$ aws dms modify-replication-instance
   --replication-instance-arn "arn:aws:dms:us-
east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV"

A client error (AccessDeniedException) occurred when calling the
 ModifyReplicationInstance
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:ModifyReplicationInstance on resource: arn:aws:dms:us-
east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV
```

Vous pouvez également spécifier des stratégies IAM qui limitent l'accès aux points de terminaison AWS DMS et aux tâches de réplication.

La stratégie suivante limite l'accès à un point de terminaison AWS DMS à l'aide de l'ARN de ce point de terminaison.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dms:*"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:dms:us-
east-1:152683116:endpoint:D6E37YBXTNHOA6XRQSZCUGX"
        }
    ]
}
```

Par exemple, les commandes suivantes échouent si la stratégie qui utilise l'ARN du point de terminaison est en vigueur.

```
$ aws dms delete-endpoint
   --endpoint-arn "arn:aws:dms:us-east-1:152683116:endpoint:D6E37YBXTNHOA6XRQSZCUGX"
```

```
A client error (AccessDeniedException) occurred when calling the DeleteEndpoint
 operation:
User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
 dms:DeleteEndpoint
on resource: arn:aws:dms:us-east-1:152683116:endpoint:D6E37YBXTNHOA6XRQSZCUGX

$ aws dms modify-endpoint
   --endpoint-arn "arn:aws:dms:us-east-1:152683116:endpoint:D6E37YBXTNHOA6XRQSZCUGX"


A client error (AccessDeniedException) occurred when calling the ModifyEndpoint
 operation:
User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
 dms:ModifyEndpoint
on resource: arn:aws:dms:us-east-1:152683116:endpoint:D6E37YBXTNHOA6XRQSZCUGX
```

La stratégie suivante limite l'accès à une tâche AWS DMS à l'aide de l'ARN de la tâche.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dms:*"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:dms:us-east-1:152683116:task:UO3YR4N47DXH3ATT4YMWOIT"
        }
    ]
}
```

Par exemple, les commandes suivantes échouent si la stratégie utilisant l'ARN de la tâche est en vigueur.

```
$ aws dms delete-replication-task
   --replication-task-arn "arn:aws:dms:us-
east-1:152683116:task:UO3YR4N47DXH3ATT4YMWOIT"
```

```
A client error (AccessDeniedException) occurred when calling the DeleteReplicationTask
 operation:
User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
 dms:DeleteReplicationTask
on resource: arn:aws:dms:us-east-1:152683116:task:UO3YR4N47DXH3ATT4YMWOIT
```

Utilisation de balises pour contrôler l'accès

AWS DMS définit un ensemble de paires clé-valeur courantes, disponibles pour être utilisées dans les stratégies définies par les clients sans qu'aucun balisage supplémentaire ne soit requis. Pour plus d'informations sur le balisage des ressources AWS DMS, consultez Balisage des ressources dans AWS Database Migration Service.

Le tableau suivant répertorie les balises standard disponibles pour une utilisation avec AWS DMS :

- aws:CurrentTime – représente la date et l'heure de la demande, ce qui permet de restreindre l'accès en fonction de critères temporels.
- aws:EpochTime : cette balise est similaire à la balise aws:CurrentTime précédente, si ce n'est que l'heure actuelle est représentée comme le nombre de secondes écoulées depuis l'époque Unix.
- aws:MultiFactorAuthPresent : balise booléenne qui indique si la demande a été signée ou non via l'authentification multifactorielle.
- aws:MultiFactorAuthAge – fournit l'accès à l'âge du jeton d'authentification multi-facteurs (en secondes).
- aws:principaltype : fournit l'accès au type de principal (utilisateur, compte, utilisateur fédéré, etc.) pour la demande actuelle.
- aws:SourceIp : représente l'adresse IP source de l'utilisateur émettant la demande.
- aws:UserAgent – fournit des informations sur l'application client demandant une ressource.
- aws:userid – fournit l'accès à l'ID de l'utilisateur émettant la demande.
- aws:username – fournit l'accès au nom de l'utilisateur émettant la demande.
- dms:InstanceClass – fournit l'accès à la taille de calcul de l'hôte ou des hôtes de l'instance de réplication.
- dms:StorageSize : fournit l'accès à la taille du volume de stockage (en Go).

Vous pouvez également définir vos propres balises. Les balises définies par le client sont de simples paires clé-valeur, conservées dans le service de balisage AWS. Vous pouvez les ajouter aux

ressources AWS DMS, y compris les instances de réplication, les points de terminaison et les tâches. Ces balises sont mises en correspondance via les instructions « conditionnelles » IAM des stratégies et sont référencées à l'aide d'une balise conditionnelle spécifique. Les clés de balise sont préfixées par « dms », le type de ressource et le préfixe « tag ». Le format de balise est le suivant.

```
dms:{resource type}-tag/{tag key}={tag value}
```

Par exemple, supposons que vous vouliez définir une stratégie permettant qu'un appel d'API ne réussisse que pour une instance de réplication contenant la balise « stage=production ». L'instruction conditionnelle suivante correspond à une ressource avec la balise donnée.

```
"Condition":
{
    "streq":
        {
            "dms:rep-tag/stage":"production"
        }
}
```

Vous devez ajouter la balise suivante à une instance de réplication correspondant à cette condition de stratégie.

```
stage production
```

En dehors des balises déjà attribuées aux ressources AWS DMS, les stratégies peuvent aussi être écrites pour limiter les clés et les valeurs de balise qui peuvent s'appliquer à une ressource donnée. Dans ce cas, le préfixe de balise est « req ».

Par exemple, la déclaration de stratégie suivante limite les balises qu'un utilisateur peut attribuer à une ressource donnée pour une liste spécifique de valeurs autorisées.

```
 "Condition":
{
    "streq":
        {
            "dms:rep-tag/stage": [ "production", "development", "testing" ]
        }
}
```

Les exemples de stratégie suivants limitent l'accès à une ressource AWS DMS en fonction des balises de la ressource.

La stratégie suivante limite l'accès à une instance de réplication où la valeur de balise est « Desktop » et la clé de balise « Env » :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dms:*"
            ],
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "dms:rep-tag/Env": [
                        "Desktop"
                    ]
                }
            }
        }
    ]
}
```

Les commandes suivantes réussissent ou échouent en fonction de la stratégie IAM qui restreint l'accès lorsque la valeur de balise est « Desktop » et la clé de balise « Env ».

```
$ aws dms list-tags-for-resource
   --resource-name arn:aws:dms:us-east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN
   --endpoint-url http://localhost:8000
{
    "TagList": [
        {
            "Value": "Desktop",
            "Key": "Env"
        }
    ]
```

```
}

$ aws dms delete-replication-instance
   --replication-instance-arn "arn:aws:dms:us-
east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN"
A client error (AccessDeniedException) occurred when calling the
 DeleteReplicationInstance
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:DeleteReplicationInstance on resource: arn:aws:dms:us-
east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN

$ aws dms modify-replication-instance
   --replication-instance-arn "arn:aws:dms:us-
east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN"

A client error (AccessDeniedException) occurred when calling the
 ModifyReplicationInstance
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:ModifyReplicationInstance on resource: arn:aws:dms:us-
east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN

$ aws dms add-tags-to-resource
   --resource-name arn:aws:dms:us-east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN
   --tags Key=CostCenter,Value=1234

A client error (AccessDeniedException) occurred when calling the AddTagsToResource
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:AddTagsToResource on resource: arn:aws:dms:us-
east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN

$ aws dms remove-tags-from-resource
   --resource-name arn:aws:dms:us-east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN
   --tag-keys Env

A client error (AccessDeniedException) occurred when calling the
 RemoveTagsFromResource
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:RemoveTagsFromResource on resource: arn:aws:dms:us-
east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN
```

La stratégie suivante limite l'accès à un point de terminaison AWS DMS où la valeur de balise est « Desktop » et la clé de balise « Env ».

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dms:*"
            ],
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "dms:endpoint-tag/Env": [
                        "Desktop"
                    ]
                }
            }
        }
    ]
}
```

Les commandes suivantes réussissent ou échouent en fonction de la stratégie IAM qui restreint l'accès lorsque la valeur de balise est « Desktop » et la clé de balise « Env ».

```
$ aws dms list-tags-for-resource
   --resource-name arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I
{
    "TagList": [
        {
            "Value": "Desktop",
            "Key": "Env"
        }
    ]
}

$ aws dms delete-endpoint
   --endpoint-arn "arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I"

A client error (AccessDeniedException) occurred when calling the DeleteEndpoint
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
```

```
dms:DeleteEndpoint on resource: arn:aws:dms:us-
east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I

$ aws dms modify-endpoint
   --endpoint-arn "arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I"


A client error (AccessDeniedException) occurred when calling the ModifyEndpoint
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:ModifyEndpoint on resource: arn:aws:dms:us-
east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I

$ aws dms add-tags-to-resource
   --resource-name arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I
   --tags Key=CostCenter,Value=1234

A client error (AccessDeniedException) occurred when calling the AddTagsToResource
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:AddTagsToResource on resource: arn:aws:dms:us-
east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I

$ aws dms remove-tags-from-resource
   --resource-name arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I
   --tag-keys Env

A client error (AccessDeniedException) occurred when calling the
 RemoveTagsFromResource
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:RemoveTagsFromResource on resource: arn:aws:dms:us-
east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I
```

La stratégie suivante limite l'accès à une tâche de réplication où la valeur de balise est « Desktop » et la clé de balise « Env ».

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dms:*"
            ],
```

```
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "dms:task-tag/Env": [
                        "Desktop"
                    ]
                }
            }
        }
    ]
}
```

Les commandes suivantes réussissent ou échouent en fonction de la stratégie IAM qui restreint l'accès lorsque la valeur de balise est « Desktop » et la clé de balise « Env ».

```
$ aws dms list-tags-for-resource
   --resource-name arn:aws:dms:us-east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3
{
    "TagList": [
        {
            "Value": "Desktop",
            "Key": "Env"
        }
    ]
}

$ aws dms delete-replication-task
   --replication-task-arn "arn:aws:dms:us-east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3"

A client error (AccessDeniedException) occurred when calling the DeleteReplicationTask
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:DeleteReplicationTask on resource: arn:aws:dms:us-
east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3

$ aws dms add-tags-to-resource
   --resource-name arn:aws:dms:us-east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3
   --tags Key=CostCenter,Value=1234

A client error (AccessDeniedException) occurred when calling the AddTagsToResource
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
```

```
dms:AddTagsToResource on resource: arn:aws:dms:us-
east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3

$ aws dms remove-tags-from-resource
   --resource-name arn:aws:dms:us-east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3
   --tag-keys Env

A client error (AccessDeniedException) occurred when calling the
 RemoveTagsFromResource
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:RemoveTagsFromResource on resource: arn:aws:dms:us-
east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3
```

Définition d'une clé de chiffrement et spécification des autorisations AWS KMS

AWS DMS chiffre le stockage utilisé par une instance de réplication et les informations de connexion de point de terminaison. Pour chiffrer le stockage utilisé par une instance de réplication, AWS DMS utilise une clé AWS Key Management Service (AWS KMS) unique pour votre compte AWS. Vous pouvez afficher et gérer cette clé avec AWS KMS. Vous pouvez utiliser la clé KMS par défaut de votre compte (`aws/dms`) ou créer une clé KMS personnalisée. Si vous disposez d'une clé KMS existante, vous pouvez également utiliser cette clé pour le chiffrement.

> **Note**
>
> Toute clé AWS KMS personnalisée ou existante que vous utilisez comme clé de chiffrement doit être une clé symétrique. AWS DMS ne prend pas en charge l'utilisation de clés de chiffrement asymétriques. Pour plus d'informations sur les clés de chiffrement symétriques et asymétriques, consultez https://docs.aws.amazon.com/kms/latest/developerguide/symmetric-asymmetric.html dans le Guide du développeur AWS Key Management Service.

La clé KMS par défaut (`aws/dms`) est créée lorsque vous lancez pour la première fois une instance de réplication, si vous n'avez pas sélectionné une clé KMS personnalisée dans la section Avancé de la page Créer une instance de réplication. Si vous utilisez la clé KMS par défaut, les seules autorisations que vous devez accorder au compte utilisateur IAM que vous utilisez pour la migration sont `kms:ListAliases` et `kms:DescribeKey`. Pour en savoir plus sur l'utilisation de la clé KMS par défaut, consultez Autorisations IAM nécessaires pour utiliser AWS DMS.

Pour utiliser une clé KMS personnalisée, vous devez attribuer des autorisations pour la clé KMS personnalisée en appliquant l'une des options suivantes :

- Ajoutez le compte utilisateur IAM utilisé pour la migration en tant qu'administrateur ou utilisateur de la clé pour la clé AWS KMS personnalisée. Cela permettra de garantir l'attribution des autorisations AWS KMS requises au compte utilisateur IAM. Cette action est effectuée en plus des autorisations IAM que vous devez attribuer au compte utilisateur IAM pour utiliser AWS DMS. Pour plus d'informations sur l'octroi d'autorisations à un utilisateur de clé, consultez Autoriser les utilisateurs de clé à utiliser la clé KMS dans le Guide du développeur AWS Key Management Service.

- Si vous ne souhaitez pas ajouter le compte utilisateur IAM en tant qu'administrateur ou utilisateur de la clé pour la clé KMS personnalisée, ajoutez les autorisations suivantes aux autorisations IAM qui doivent être accordées au compte utilisateur IAM pour utiliser AWS DMS.

```
{
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases",
                "kms:DescribeKey",
                "kms:CreateGrant",
                "kms:Encrypt",
                "kms:ReEncrypt*"
            ],
            "Resource": "*"
        },
```

AWS DMS fonctionne également avec les alias de clé KMS. Pour plus d'informations sur la création de vos propres clés AWS KMS et sur la configuration de l'accès des utilisateurs à une clé KMS, consultez le Guide du développeur AWS KMS.

Si vous ne précisez pas d'identifiant de clé KMS, AWS DMS utilise votre clé de chiffrement par défaut. AWS KMS crée la clé de chiffrement par défaut pour AWS DMS, pour votre compte AWS. Votre compte AWS a une clé de chiffrement par défaut différente pour chaque région AWS.

Pour gérer les clés AWS KMS utilisées pour le chiffrement et le déchiffrement de vos ressources AWS DMS, utilisez le AWS Key Management Service. AWS KMS combine du matériel et des logiciels sécurisés et hautement disponibles pour fournir un système de gestion de clés à l'échelle du cloud. Grâce à AWS KMS, vous pouvez créer des clés de chiffrement et définir les stratégies qui contrôlent la manière dont ces clés peuvent être utilisées.

Vous pouvez trouver AWS KMS dans la AWS Management Console.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Choisissez l'une des options suivantes pour utiliser les clés AWS KMS :

- Pour afficher les clés de votre compte qu'AWS crée et gère pour vous, dans le panneau de navigation, choisissez Clés gérées par AWS.
- Pour afficher les clés de votre compte que vous créez et gérez vous-même, dans le volet de navigation, choisissez Clés gérées par le client.

AWS KMS prend en charge AWS CloudTrail pour vous permettre d'effectuer l'audit de l'utilisation des clés et vérifier que cette utilisation est appropriée. Vous pouvez utiliser vos clés AWS KMS avec AWS DMS et les services AWS pris en charge tels qu'Amazon RDS, Amazon S3, Amazon Redshift et Amazon EBS.

Vous pouvez également créer des clés AWS DMS personnalisées pour chiffrer spécifiquement des données cibles pour les points de terminaison AWS KMS suivants :

- Amazon Redshift : pour plus d'informations, consultez Création et utilisation de clés AWS KMS pour chiffrer les données cibles Amazon Redshift.
- Amazon S3 : pour plus d'informations, consultez Création de clés AWS KMS pour chiffrer des objets cibles Amazon S3.

Une fois que vous avez créé vos ressources AWS DMS avec une clé KMS, vous ne pouvez pas modifier la clé de chiffrement pour ces ressources. Prenez soin de déterminer vos besoins en termes de clés de chiffrement avant de créer vos ressources AWS DMS.

Sécurité de réseau pour AWS Database Migration Service

Les exigences de sécurité pour le réseau que vous créez lors de l'utilisation d'AWS Database Migration Service dépendent de la façon dont vous configurez le réseau. Les règles générales pour la sécurité du réseau pour AWS DMS sont les suivantes :

- L'instance de réplication doit avoir accès aux points de terminaison source et cible. Le groupe de sécurité pour l'instance de réplication doit disposer d'ACL réseau ou de règles qui autorisent le trafic sortant à partir de l'instance sur le port de base de données vers les points de terminaison de base de données.
- Les points de terminaison de base de données doivent inclure des ACL réseau et des règles de groupe de sécurité qui autorisent l'accès entrant à partir de l'instance de réplication. Vous pouvez atteindre cet objectif à l'aide du groupe de sécurité de l'instance de réplication, l'adresse IP privée, l'adresse IP publique ou l'adresse publique de la passerelle NAT, selon votre configuration.
- Si votre réseau utilise un tunnel VPN, l'instance Amazon EC2 qui agit comme la passerelle NAT doit utiliser un groupe de sécurité possédant des règles qui permettent à l'instance de réplication d'envoyer le trafic par son intermédiaire.

Par défaut, le groupe de sécurité VPC utilisé par l'instance de réplication AWS DMS possède des règles qui autorisent le trafic sortant à 0.0.0.0/0 sur tous les ports. Si vous modifiez ce groupe de sécurité ou que vous utilisez votre propre groupe de sécurité, le trafic sortant doit, au minimum, être autorisé sur les points de terminaison sources et cibles des ports de base de données respectifs.

Les configurations de réseau que vous pouvez utiliser pour la migration de base de données chaque requièrent toutes des normes de sécurité spécifiques :

- Configuration avec tous les composants de migration de base de données dans un VPC : le groupe de sécurité utilisé par les points de terminaison doit autoriser le trafic entrant sur le port de base de données à partir de l'instance de réplication. Veillez que le groupe de sécurité utilisé par l'instance de réplication dispose du trafic entrant vers les points de terminaison, ou créez une règle dans le groupe de sécurité utilisé par les points de terminaison qui autorise l'adresse IP privée de l'accès à la réplication d'instance.
- Configuration avec plusieurs VPC : le groupe de sécurité utilisé par l'instance de réplication doit avoir une règle pour la plage VPC et le port de base de données sur la base de données.
- Configuration d'un réseau vers un VPC à l'aide AWS Direct Connect d'un VPN : tunnel VPN autorisant le trafic à accéder du VPC à un VPN sur site. Dans cette configuration, le VPC comporte

une règle de routage qui envoie le trafic destiné à une adresse IP spécifique ou une plage vers un hôte qui peut acheminer le trafic du VPC vers le VPN sur site. Dans ce cas, l'hôte NAT inclut ses propres paramètres de groupe de sécurité qui doivent autoriser le trafic provenant de l'adresse IP privée ou du groupe de sécurité de l'instance de réplication vers l'instance NAT.

- Configuration d'un réseau à un VPC avec Internet : le groupe de sécurité du VPC doit inclure des règles de routage qui envoient le trafic non destiné au VPC à la passerelle Internet. Dans cette configuration, la connexion au point de terminaison semble provenir de l'adresse IP publique sur l'instance de réplication.
- Configuration avec une instance de base de données RDS ne figurant pas dans un VPC vers une instance de base de données dans un VPC en utilisant ClassicLink : quand l'instance de base de données Amazon RDS source ou cible ne se trouve pas dans un VPC et ne partage pas un groupe de sécurité avec le VPC où se trouve l'instance de réplication, vous pouvez configurer un serveur proxy et utiliser ClassicLink pour connecter les bases de données source et cible.
- Le point de terminaison source est en dehors du VPC utilisé par l'instance de réplication et utilise une passerelle NAT : vous pouvez configurer une passerelle de traduction d'adresses réseau (NAT) en utilisant une seule adresse IP Elastic liée à une seule interface réseau Elastic. Celle-ci reçoit alors un identificateur NAT (nat-#####). Si le VPC comporte une route par défaut pour cette passerelle NAT au lieu de la passerelle Internet, l'instance de réplication apparaît pour contacter le point de terminaison de base de données à l'aide de l'adresse IP publique de la passerelle Internet. Dans ce cas, le trafic entrant vers le point de terminaison de base de données en dehors du VPC doit autoriser le trafic entrant à partir de l'adresse NAT au lieu de l'adresse IP publique de l'instance de réplication.
- Points de terminaison de VPC pour les moteurs non-SGBDR : AWS DMS ne prend pas en charge les points de terminaison de VPC pour les moteurs non-SGBDR.

Utilisation du protocole SSL avec AWS Database Migration Service

Vous pouvez chiffrer les connexions pour les points de terminaison source et cible grâce au protocole SSL (Secure Sockets Layer). Pour ce faire, vous pouvez utiliser la console de gestion AWS DMS ou l'API AWS DMS pour attribuer un certificat à un point de terminaison. Vous pouvez également utiliser la console AWS DMS pour gérer vos certificats.

Toutes les bases de données n'utilisent pas SSL de la même manière. Amazon Aurora MySQL-Compatible Edition utilise le nom du serveur, le point de terminaison de l'instance principale dans le cluster, en tant que point de terminaison pour SSL. Un point de terminaison Amazon Redshift utilise déjà une connexion SSL et ne requiert pas de connexion SSL configurée par AWS DMS. Un point de terminaison Oracle requiert des étapes supplémentaires ; pour plus d'informations, consultez Prise en charge de SSL pour un point de terminaison Oracle.

Rubriques

- Limitations sur l'utilisation de SSL avec AWS DMS
- Gestion de certificats
- Activation de SSL pour un point de terminaison compatible MySQL, PostgreSQL ou SQL Server

Pour attribuer un certificat à un point de terminaison, vous fournissez le certificat racine ou la chaîne de certificats CA intermédiaires menant à la racine (sous la forme d'un ensemble de certificats) qui a été utilisée pour signer le certificat SSL du serveur qui est déployé sur votre point de terminaison. Les certificats sont acceptés uniquement sous la forme de fichiers X509 au format PEM. Lorsque vous importez un certificat, vous recevez un ARN (Amazon Resource Name), que vous pouvez utiliser pour spécifier ce certificat pour un point de terminaison. Si vous utilisez Amazon RDS, vous pouvez télécharger l'autorité de certification racine et l'ensemble de certificats fourni dans le fichier `rds-combined-ca-bundle.pem` hébergé par Amazon RDS. Pour plus d'informations sur le téléchargement de ce fichier, consultez Utilisation de SSL/TLS pour chiffrer une connexion à une instance de base de données dans le Guide de l'utilisateur Amazon RDS.

Vous avez le choix parmi plusieurs modes SSL à utiliser pour vérifier le certificat SSL.

- none : la connexion n'est pas chiffrée. Cette option n'est pas sécurisée, mais elle nécessite moins de ressources.
- require : la connexion est chiffrée à l'aide du protocole SSL (TLS), mais aucune vérification d'autorité de certification n'est effectuée. Cette option est plus sécurisée, mais elle nécessite plus de ressources.

- verify-ca : la connexion est chiffrée. Cette option est plus sécurisée, mais elle nécessite plus de ressources. Cette option vérifie le certificat de serveur.
- verify-full : la connexion est chiffrée. Cette option est plus sécurisée, mais elle nécessite plus de ressources. Cette option vérifie le certificat de serveur et vérifie que le nom d'hôte du serveur correspond à l'attribut de nom d'hôte pour le certificat.

Les modes SSL ne fonctionnent pas tous avec tous les points de terminaison de base de données. Le tableau suivant présente les modes SSL pris en charge pour chaque moteur de base de données.

Moteur de base de données	Aucun(e)	require	verify-ca	verify-full
MySQL/MariaDB/ Amazon Aurora MySQL	Par défaut	Non pris en charge	Pris en charge	Pris en charge
Microsoft SQL Server	Par défaut	Pris en charge	Non pris en charge	Pris en charge
PostgreSQL	Par défaut	Pris en charge	Pris en charge	Pris en charge
Amazon Redshift	Par défaut	SSL non activé	SSL non activé	SSL non activé
Oracle	Par défaut	Non pris en charge	Pris en charge	Non pris en charge
SAP ASE	Par défaut	SSL non activé	SSL non activé	Pris en charge
MongoDB	Par défaut	Pris en charge	Non pris en charge	Pris en charge
Db2 LUW	Par défaut	Non pris en charge	Pris en charge	Non pris en charge
Db2 pour z/OS	Par défaut	Non pris en charge	Pris en charge	Non pris en charge

> **Note**
>
> L'option Mode SSL de la console DMS ou de l'API ne s'applique pas à certains services de streaming de données et NoSQL tels que Kinesis et DynamoDB. Ils sont sécurisés par défaut, si bien que DMS indique que le paramètre du mode SSL a pour valeur none (SSL Mode=None). Vous n'avez pas besoin de fournir de configuration supplémentaire pour que votre point de terminaison utilise le protocole SSL. Par exemple, lorsque Kinesis est utilisé comme point de terminaison cible, il est sécurisé par défaut. Tous les appels d'API à Kinesis utilisent le protocole SSL. Il n'est donc pas nécessaire d'ajouter une option SSL supplémentaire dans le point de terminaison DMS. Vous pouvez placer et récupérer des données en toute sécurité via des points de terminaison SSL à l'aide du protocole HTTPS, utilisé par DMS par défaut lors de la connexion à un flux de données Kinesis.

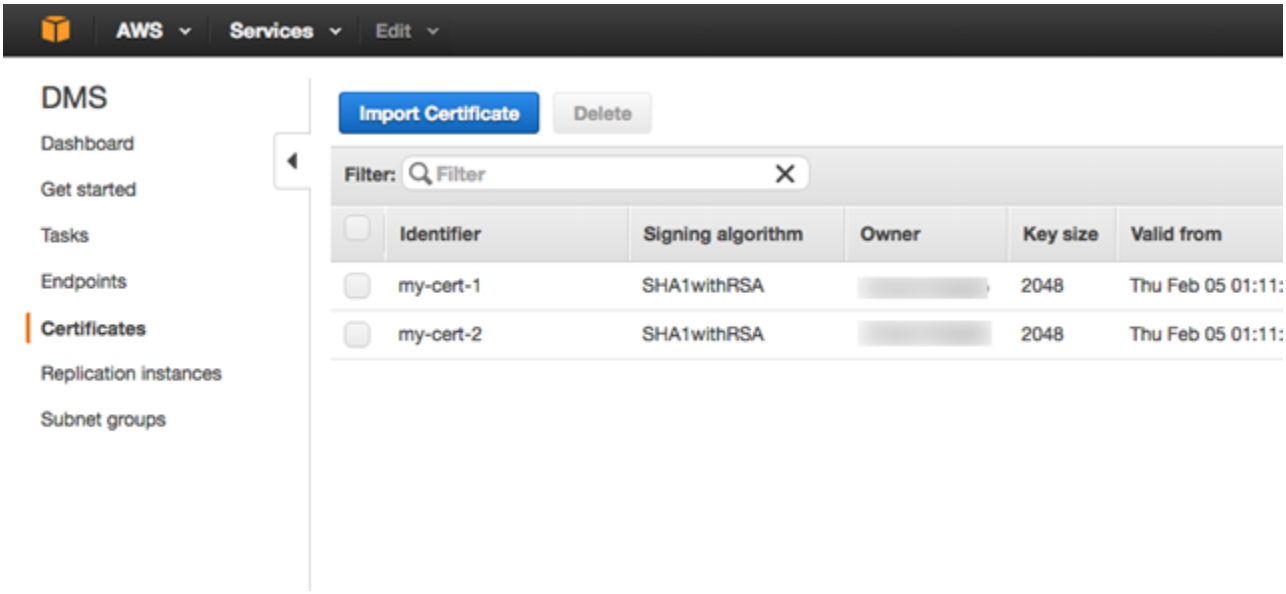
Limitations sur l'utilisation de SSL avec AWS DMS

Voici les limitations relatives à l'utilisation de SSL avec AWS DMS :

- Les connexions SSL aux points de terminaison cibles Amazon Redshift ne sont pas prises en charge. AWS DMS utilise un compartiment Amazon S3 pour transférer les données vers la base de données Amazon Redshift. Cette transmission est chiffrée par Amazon Redshift par défaut.
- Des délais d'attente SQL peuvent se produire lors de l'exécution de tâches de capture des changements de données (CDC) avec les points de terminaison Oracle compatibles SSL. Si vous rencontrez un problème, dans lequel les compteurs CDC ne reflètent pas les nombres attendus, affectez une valeur inférieure au paramètre `MinimumTransactionSize` dans la section `ChangeProcessingTuning` des paramètres de tâche. La valeur la plus faible possible correspond à 100. Pour plus d'informations sur le paramètre `MinimumTransactionSize`, consultez Paramètres de réglage du traitement des modifications.
- Vous ne pouvez importer les certificats que dans les formats .pem et .sso (portefeuille Oracle).
- Dans certains cas, le certificat SSL de votre serveur peut être signé par une autorité de certification intermédiaire. Si tel est le cas, assurez-vous que toute la chaîne de certificats allant de l'autorité de certification intermédiaire à l'autorité de certification racine est importée en tant que fichier .pem unique.
- Si vous utilisez des certificats auto-signés sur votre serveur, sélectionnez require comme mode SSL. Le mode SSL require fait confiance implicitement au certificat SSL du serveur et ne tente pas de s'assurer que le certificat a été signé par une autorité de certification.

Gestion de certificats

Vous pouvez utiliser la console DMS pour afficher et gérer vos certificats SSL. Vous pouvez également importer vos certificats à l'aide de la console DMS.



Activation de SSL pour un point de terminaison compatible MySQL, PostgreSQL ou SQL Server

Vous pouvez ajouter une connexion SSL à un point de terminaison nouvellement créé ou à un point de terminaison existant.

Pour créer un point de terminaison AWS DMS avec SSL

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS DMS à l'adresse https://console.aws.amazon.com/dms/v2/.

   Si vous êtes connecté en tant qu'utilisateur AWS Identity and Access Management (IAM), vous devez détenir les autorisations appropriées pour accéder à AWS DMS. Pour plus d'informations sur les autorisations requises pour la migration de base de données, consultez Autorisations IAM nécessaires pour utiliser AWS DMS.
2. Dans le volet de navigation, choisissez Certificats.
3. Sélectionnez Importer un certificat.
4. Téléchargez le certificat que vous voulez utiliser pour chiffrer la connexion à un point de terminaison.

> **Note**
>
> Vous pouvez également télécharger un certificat à l'aide de la console AWS DMS lorsque vous créez ou modifiez un point de terminaison en sélectionnant Ajouter un certificat CA sur la page Créer un point de terminaison de base de données. Pour Aurora sans serveur en tant que cible, obtenez le certificat mentionné dans Utilisation de TLS/SSL avec Aurora sans serveur.

5. Créez un point de terminaison comme décrit dans la page Étape 2 : Spécifier les points de terminaison sources et cibles

Pour modifier un point de terminaison AWS DMS existant afin d'utiliser le protocole SSL :

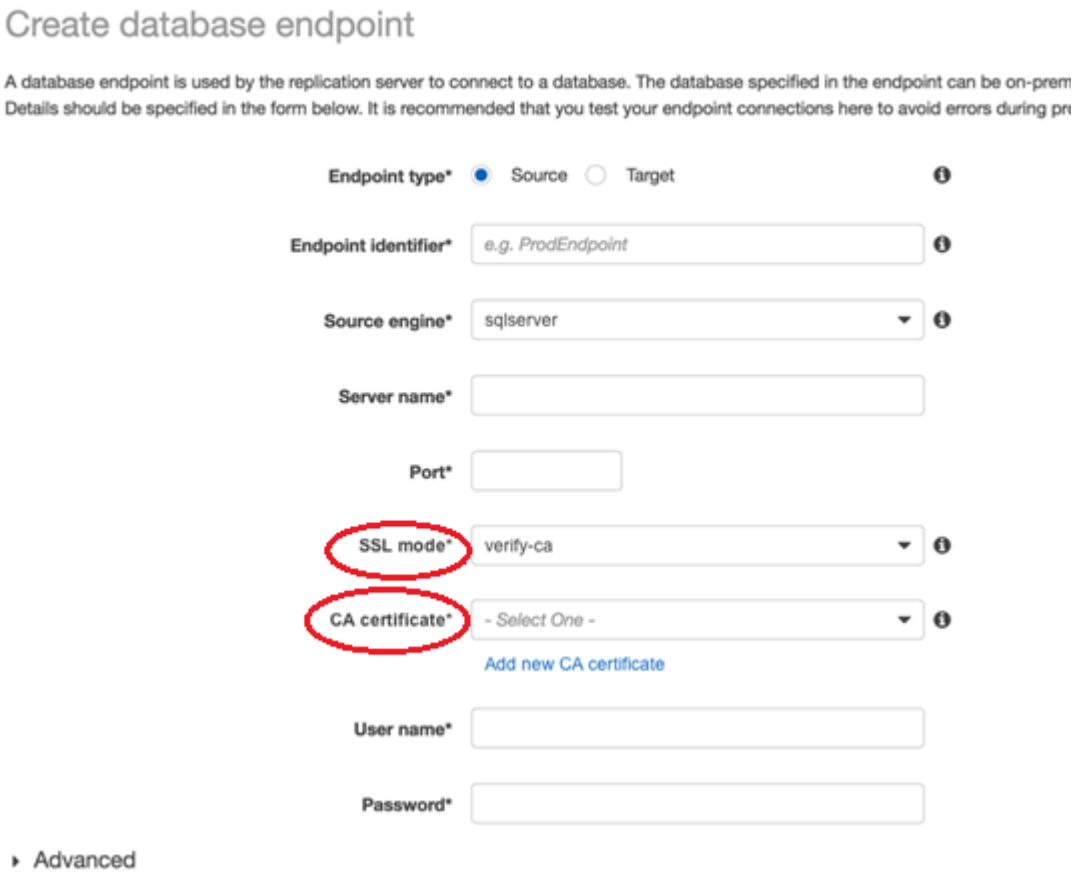
1. Connectez-vous à la AWS Management Console et ouvrez la console AWS DMS à l'adresse https://console.aws.amazon.com/dms/v2/.

   Si vous êtes connecté en tant qu'utilisateur IAM, vous devez détenir les autorisations appropriées pour accéder à AWS DMS. Pour plus d'informations sur les autorisations requises pour la migration de base de données, consultez Autorisations IAM nécessaires pour utiliser AWS DMS.
2. Dans le volet de navigation, choisissez Certificats.
3. Sélectionnez Importer un certificat.
4. Téléchargez le certificat que vous voulez utiliser pour chiffrer la connexion à un point de terminaison.

> **Note**
>
> Vous pouvez également télécharger un certificat à l'aide de la console AWS DMS lorsque vous créez ou modifiez un point de terminaison en sélectionnant Ajouter un certificat CA sur la page Créer un point de terminaison de base de données.

5. Dans le volet de navigation, sélectionnez Points de terminaison, sélectionnez le point de terminaison à modifier, puis Modifier.
6. Choisissez une valeur pour le mode SSL.

Si vous choisissez le mode verify-ca ou verify-full, spécifiez le certificat que vous souhaitez utiliser pour le certificat d'autorité de certification, comme indiqué ci-dessous.



7. Sélectionnez Modifier.
8. Lorsque le point de terminaison a été modifié, sélectionnez-le et choisissez Tester la connexion pour déterminer si la connexion SSL fonctionne.

Lorsque vous avez créé vos points de terminaison source et cible, créez une tâche qui utilise ces points de terminaison. Pour plus d'informations sur la création d'une tâche, consultez Étape 3 : Créer une tâche et migrer les données.

Modification du mot de passe de base de données

Dans la plupart des cas, la modification du mot de passe de base de données pour votre point de terminaison source ou cible est un processus simple. Si vous avez besoin modifier le mot de passe de base de données pour un point de terminaison que vous utilisez actuellement dans une tâche de réplication ou de migration, le processus est un peu plus complexe. La procédure suivante montre comment procéder.

Pour modifier le mot de passe de base de données pour un point de terminaison dans une tâche de réplication ou de migration

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS DMS à l'adresse https://console.aws.amazon.com/dms/v2/.

   Si vous êtes connecté en tant qu'utilisateur IAM, vous devez détenir les autorisations appropriées pour accéder à AWS DMS. Pour plus d'informations sur les autorisations requises, consultez Autorisations IAM nécessaires pour utiliser AWS DMS.
2. Dans le volet de navigation, choisissez, Tâches de migration de base de données.
3. Choisissez la tâche qui utilise le point de terminaison pour lequel vous voulez modifier le mot de passe de base de données, puis choisissez Arrêter.
4. Pendant que la tâche est en cours d'arrêt, vous pouvez modifier le mot de passe de la base de données pour le point de terminaison à l'aide des outils natifs que vous utilisez pour gérer la base de données.
5. Revenez à la console de gestion DMS et choisissez Points de terminaison dans le volet de navigation.
6. Choisissez le point de terminaison pour la base de données dont vous avez modifié le mot de passe, puis sélectionnez Modifier.
7. Tapez le nouveau mot de passe dans la zone Mot de passe, puis choisissez Modifier.
8. Dans le volet de navigation, choisissez, Tâches de migration de base de données.
9. Choisissez la tâche que vous avez arrêtée précédemment, puis sélectionnez Arrêter/Reprendre.
10. Choisissez Démarrer ou Reprendre selon la façon dont vous voulez continuer la tâche, puis choisissez Démarrer tâche.

Quotas pour AWS Database Migration Service

Vous trouverez ci-après les quotas de ressources et les contraintes d'attribution de noms pour AWS Database Migration Service (AWS DMS).

La taille maximale d'une base de données qu'AWS DMS peut migrer dépend d'un certain nombre de facteurs. Ceux-ci incluent votre environnement source, la distribution des données dans la base de données source et le niveau de sollicitation de votre système source.

Le meilleur moyen de déterminer si le système est un bon candidat pour AWS DMS est de le tester. Commencez lentement afin de pouvoir affiner la configuration, puis ajoutez des objets complexes. Enfin, tentez un chargement complet à titre de test.

Quotas de ressources pour AWS Database Migration Service

Chaque compte AWS a des quotas pour chaque région AWS sur le nombre de ressources AWS DMS pouvant être créées. Une fois qu'un quota de ressource a été atteint, les appels supplémentaires pour créer cette ressource échouent avec une exception.

Le tableau suivant répertorie les ressources AWS DMS et leurs quotas pour chaque région AWS.

Ressource	Quota par défaut
Limitation des demandes d'API	200 demandes au maximum par seconde
Taux de rafraîchissement des demandes d'API	8 demandes par seconde
Instances de réplication par compte d'utilisateur	60
Quantité totale de stockage pour une instance de réplication	30 000 Go
Abonnements aux événements par compte d'utilisateur	60
Groupes de sous-réseaux de réplication par compte d'utilisateur	60

Ressource	Quota par défaut
Sous-réseaux par groupe de sous-réseaux de réplication	60
Points de terminaison par compte d'utilisateur	1 000
Points de terminaison par instance de réplication	100
Tâches par compte d'utilisateur	600
Tâches par instance de réplication	200
Certificats par compte d'utilisateur	100
Fournisseurs de données par compte d'utilisateur	1 000
Profils d'instances par compte d'utilisateur	60
Projets de migration par compte d'utilisateur	10
Collecteurs de données DMS par compte d'utilisateur	10
Recommandations cibles générées en une seule fois	100
Nombre de fichiers que le collecteur de données DMS peut charger par heure	500
Migrations de données homogènes par compte d'utilisateur	600
Migrations de données homogènes exécutées en une seule fois	100
Migrations de données homogènes par projet de migration	10
Réplications sans serveur	100

Pour plus d'informations sur le quota de limitation des demandes d'API et le taux de rafraîchissement, consultez Comprendre la limitation des demandes d'API.

Le quota de 30 000 Go de stockage s'applique à toutes vos instances de réplication AWS DMS dans une région AWS donnée. Ce stockage est utilisé pour mettre en cache les modifications si une cible ne peut satisfaire les exigences de la source et pour stocker les informations de journalisation.

Comprendre la limitation des demandes d'API

AWS DMS prend en charge un quota de demandes d'API variable mais maximal de 200 appels d'API par seconde. En d'autres termes, une limitation s'applique à vos demandes d'API lorsqu'elles dépassent ce taux. Vous pouvez également être limité à un nombre réduit d'appels d'API par seconde, selon le temps nécessaire à AWS DMS pour actualiser votre quota avant que vous effectuiez une autre demande d'API. Ce quota s'applique à la fois quand vous effectuez des appels d'API directement et quand ils sont effectués en votre nom dans le cadre de l'utilisation de la console de gestion AWS DMS.

Pour comprendre le fonctionnement de la limitation des demandes d'API, il est utile d'imaginer qu'AWS DMS gère un seau à jetons pour effectuer le suivi de vos demandes d'API. Dans ce scénario, chaque jeton du seau vous permet d'effectuer un appel d'API unique. Vous ne pouvez pas avoir plus de 200 jetons dans le seau à la fois. Lorsque vous effectuez un appel d'API, AWS DMS retire un jeton du seau. Si vous effectuez 200 appels d'API en moins d'une seconde, votre seau est vide et toute tentative d'effectuer un autre appel d'API échoue. Pour chaque seconde pendant laquelle vous n'effectuez pas d'appel d'API, AWS DMS ajoute 8 jetons dans le seau, jusqu'au maximum de 200 jetons. Il s'agit du taux de rafraîchissement des demandes d'API d'AWS DMS. À tout moment après la limitation, lorsque des jetons sont ajoutés à votre seau, vous pouvez effectuer autant d'appels d'API supplémentaires qu'il y a de jetons disponibles jusqu'à ce que vos appels soient à nouveau limités.

Si vous utilisez AWS CLI pour exécuter des appels d'API qui sont limités, AWS DMS renvoie une erreur comme celle-ci :

```
An error occurred (ThrottlingException) when calling the AwsDmsApiCall operation
 (reached max retries: 2): Rate exceeded
```

Dans ce cas, *AwsDmsApiCall* est le nom de l'opération d'API AWS DMS qui a été limitée, par exemple, `DescribeTableStatistics`. Vous pouvez ensuite retenter l'appel ou en effectuer un autre après un délai suffisant pour éviter la limitation.

> **Note**
>
> Contrairement à la limitation des demandes d'API gérée par d'autres services, tels qu'Amazon EC2, vous ne pouvez pas demander une augmentation des quotas de limitation des demandes d'API gérés par AWS DMS.

Résolution des problèmes liés aux tâches de migration AWS Database Migration Service

Vous trouverez ci-dessous des rubriques relatives à la résolution des problèmes liés à AWS Database Migration Service (AWS DMS). Ces rubriques peuvent vous aider à résoudre les problèmes courants en utilisant à la fois AWS DMS et les bases de données de point de terminaison sélectionnées.

Si vous avez ouvert un dossier AWS Support, votre ingénieur support peut éventuellement identifier un problème potentiel lié à l'une de vos configurations de la base de données des points de terminaison. Votre ingénieur peut également vous demander d'exécuter un script d'assistance pour renvoyer des informations de diagnostic concernant la base de données. Pour plus de détails sur le téléchargement, l'exécution et le chargement des informations de diagnostic à partir de ce type de script d'assistance, consultez Utilisation de scripts d'assistance au diagnostic dans AWS DMS.

À des fins de résolution des problèmes, AWS DMS collecte les fichiers de trace et de vidage dans l'instance de réplication. Vous pouvez fournir ces fichiers au AWS Support en cas de problème nécessitant un dépannage. Par défaut, DMS purge les fichiers de trace et de vidage datant de plus de trente jours. Pour désactiver la collecte de fichiers de trace et de vidage, ouvrez un dossier auprès du AWS Support.

Rubriques

- Les tâches de migration s'exécutent lentement
- La barre de statut des tâches ne bouge pas
- La tâche est terminée mais rien n'a été migré
- Des clés étrangères et des index secondaires sont manquants
- AWS DMSne crée pas de CloudWatch journaux
- Des problèmes se produisent lors de la connexion à Amazon RDS
- Des problèmes de réseau surviennent
- La capture des données modifiées est bloquée après le chargement complet
- Erreurs de violation de clé primaire lors du redémarrage d'une tâche
- Échec du chargement initial d'un schéma
- Échec des tâches avec une erreur inconnue
- Le redémarrage d'une tâche charge les tables dès le début

- Le nombre de tables par tâche pose problème
- Les tâches échouent quand une clé primaire est créée sur une colonne LOB
- Des doublons d'enregistrements apparaissent sur une table cible sans clé primaire
- Échec des points de terminaison sources dans la plage IP réservée
- Les horodatages sont brouillés dans les requêtes Amazon Athena
- Résolution de problèmes avec Oracle
- Résolution des problèmes liés à MySQL
- Résolution des problèmes liés à PostgreSQL
- Résolution des problèmes liés à Microsoft SQL Server
- Résolution des problèmes liés à Amazon Redshift
- Résolution des problèmes liés à Amazon Aurora MySQL
- Résolution des problèmes liés à SAP ASE
- Résolution des problèmes liés à IBM Db2
- Résolution des problèmes de latence dans AWS Database Migration Service
- Utilisation de scripts d'assistance au diagnostic dans AWS DMS
- Utilisation de l'AMI d'assistance au diagnostic AWS DMS

Les tâches de migration s'exécutent lentement

Plusieurs problèmes peuvent entraîner la lenteur d'exécution d'une tâche de migration, ou pour les tâches suivantes, une lenteur d'exécution supérieure à celle de la tâche initiale.

La raison la plus courante de la lenteur d'exécution d'une tâche de migration est le fait que des ressources inappropriées soient allouées à l'instance de réplication AWS DMS. Pour vous assurer que votre instance dispose de suffisamment de ressources pour les tâches que vous exécutez, vérifiez l'utilisation que fait votre instance de réplication du CPU, de la mémoire, des fichiers d'échange et des IOPS. Par exemple, plusieurs tâches avec Amazon Redshift comme point de terminaison utilisent un volume important d'E/S. Vous pouvez augmenter le nombre d'E/S par seconde pour votre instance de réplication ou fractionner vos tâches sur plusieurs instances de réplication pour une migration plus efficace.

Pour plus d'informations sur la détermination de la taille de votre instance de réplication, consultez Sélection de la meilleure taille pour une instance de réplication.

Vous pouvez augmenter la vitesse d'une charge de migration initiale en procédant comme suit :

- Si votre cible est une instance de base de données Amazon RDS, vérifiez que l'option Multi-AZ n'est pas activée pour l'instance de base de données cible.
- Désactivez les sauvegardes ou la journalisation automatiques sur la base de données cible au cours du chargement et réactivez ces fonctionnalités une fois la migration terminée.
- Si cette fonctionnalité est disponible sur votre cible, utilisez les IOPS provisionnés.
- Si vos données de migration contiennent des LOB, vérifiez que la tâche est optimisée pour la migration d'objets LOB. Pour plus d'informations sur l'optimisation pour les objets LOB, consultez Paramètres de métadonnées des tâches cibles.

La barre de statut des tâches ne bouge pas

La barre d'état des tâches donne une estimation de la progression de la tâche. La qualité de cette estimation dépend de la qualité des statistiques de table de la base de données source ; meilleures sont les statistiques de table, plus précise sera l'estimation.

Pour une tâche avec une seule table ne disposant pas de statistiques de lignes estimées, AWS DMS ne peut pas fournir une estimation complète de pourcentage. Dans ce cas, utilisez l'état de la tâche et l'indication des lignes chargées pour confirmer que la tâche est bien en cours d'exécution et de progression.

La tâche est terminée mais rien n'a été migré

Procédez comme suit si rien n'a été migré une fois votre tâche terminée.

- Vérifiez si l'utilisateur qui a créé le point de terminaison dispose d'un accès en lecture à la table que vous souhaitez migrer.
- Vérifiez si l'objet que vous souhaitez migrer est une table. S'il s'agit d'une vue, mettez à jour les mappages de tables et affectez à l'option object-locator la valeur « view » ou « all ». Pour plus d'informations, consultez Spécification des règles de sélection de table et de transformation à partir de la console.

Des clés étrangères et des index secondaires sont manquants

AWS DMS crée des tables, des clés primaires et, dans certains cas, des index uniques, mais il ne crée pas d'autres objets qui ne sont pas requis pour migrer efficacement les données de la source.

Par exemple, il ne crée pas d'index secondaires, de contraintes de clés non primaires ou de valeurs de données par défaut.

Pour migrer des objets secondaires à partir de votre base de données, utilisez les outils natifs de la base de données si vous effectuez une migration vers le même moteur de base de données que votre base de données source. Utilisez l'AWS Schema Conversion Tool (AWS SCT) si vous effectuez une migration vers un moteur de base de données autre que celui utilisé par la base de données source pour migrer des objets secondaires.

AWS DMSne crée pas de CloudWatch journaux

Si votre tâche de réplication ne crée pas de CloudWatch journaux, assurez-vous que votre compte possède le `dms-cloudwatch-logs-role` rôle. Si ce rôle n'est pas présent, procédez comme suit pour le créer :

1. Connectez-vous à l'AWS Management Console et ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Choisissez l'onglet Rôles. Sélectionnez Créer un rôle.
3. Dans la section Sélectionner le type d'entité de confiance, choisissez Service AWS.
4. Dans la section Choisir un cas d'utilisation, choisissez DMS.
5. Sélectionnez Next: Permissions (Étape suivante : autorisations).
6. Entrez **`AmazonDMSCloudWatchLogsRole`** dans le champ de recherche et cochez la case à côté d'AmazonDMS. CloudWatchLogsRole Cela donne AWS DMS des autorisations d'accès CloudWatch.
7. Choisissez Suivant : Balises.
8. Choisissez Suivant : Vérification.
9. Entrez **`dms-cloudwatch-logs-role`** pour Nom du rôle. Ce nom est sensible à la casse.
10. Sélectionnez Créer un rôle.

Des problèmes se produisent lors de la connexion à Amazon RDS

Il peut y avoir plusieurs raisons pour lesquelles vous ne pouvez pas vous connecter à une instance de base de données Amazon RDS que vous définissez en tant que source ou cible. Voici quelques points à vérifier :

- Vérifiez que la combinaison du nom d'utilisateur et du mot de passe est correcte.

- Vérifiez que la valeur de point de terminaison affichée dans la console Amazon RDS pour l'instance est identique à l'identifiant de point de terminaison que vous avez utilisé pour créer le point de terminaison AWS DMS.
- Vérifiez que la valeur de port affichée dans la console Amazon RDS pour l'instance est la même que le port affecté au point de terminaison AWS DMS.
- Vérifiez que le groupe de sécurité assigné à l'instance de base de données Amazon RDS permet des connexions à partir de l'instance de réplication AWS DMS.
- Si l'instance de réplication AWS DMS et l'instance de base de données Amazon RDS ne se trouvent pas dans le même cloud privé virtuel (VPC), vérifiez que l'instance de base de données est accessible publiquement.

Message d'erreur : Incorrect thread connection string: incorrect thread value 0

Cette erreur peut souvent se produire lorsque vous testez la connexion vers un point de terminaison. Cette erreur indique la présence d'une erreur dans la chaîne de connexion. Un espace situé après l'adresse IP de l'hôte en est un exemple. Un caractère incorrect copié dans la chaîne de connexion est un autre exemple.

Des problèmes de réseau surviennent

Les problèmes de mise en réseau les plus répandus portent sur le groupe de sécurité VPC utilisé par l'instance de réplication AWS DMS. Par défaut, ce groupe de sécurité possède des règles qui autorisent le trafic sortant à 0.0.0.0/0 sur tous les ports. Dans de nombreux cas, vous modifiez ce groupe de sécurité ou vous utilisez votre propre groupe de sécurité. Si tel est le cas, assurez-vous au moins que les points de terminaison sources et cibles disposent d'une sortie sur leurs ports de base de données respectifs.

Les autres problèmes liés à la configuration peuvent inclure les suivants :

- L'instance de réplication et les points de terminaison sources et cibles se trouvent dans le même VPC : le groupe de sécurité utilisé par les points de terminaison doit autoriser le trafic entrant sur le port de base de données à partir de l'instance de réplication. Veillez à ce que le groupe de sécurité utilisé par l'instance de réplication ait accès aux points de terminaison. Vous pouvez également créer une règle dans le groupe de sécurité utilisé par les points de terminaison qui autorise l'accès à l'adresse IP privée de l'instance de réplication.

- Le point de terminaison source se trouve en dehors du VPC utilisé par l'instance de réplication (avec une passerelle Internet) : le groupe de sécurité du VPC doit inclure des règles de routage qui envoient le trafic non destiné au VPC vers la passerelle Internet. Dans cette configuration, la connexion au point de terminaison semble provenir de l'adresse IP publique sur l'instance de réplication.
- Le point de terminaison source est en dehors du VPC utilisé par l'instance de réplication (avec une passerelle NAT) : vous pouvez configurer une passerelle de traduction d'adresses réseau (NAT) en utilisant une adresse IP Elastic unique liée à une interface réseau Elastic unique. Cette passerelle NAT reçoit un identifiant NAT (nat-######).

  Dans certains cas, le VPC inclut une route par défaut vers cette passerelle NAT au lieu de la passerelle Internet. Dans de tels cas, l'instance de réplication semble plutôt contacter le point de terminaison de base de données en utilisant l'adresse IP publique de la passerelle NAT. Ici, le trafic entrant vers le point de terminaison de base de données en dehors du VPC doit autoriser le trafic entrant à partir de l'adresse NAT au lieu de l'adresse IP publique de l'instance de réplication.

Pour en savoir plus sur l'utilisation de votre propre serveur de noms sur site, consultez Utilisation de votre propre serveur de noms sur site.

La capture des données modifiées est bloquée après le chargement complet

Il se peut que les modifications de réplication soient lentes ou bloquées après une migration de chargement complet si plusieurs paramètres AWS DMS sont conflit les uns avec les autres.

Supposons, par exemple, que le paramètre Mode de préparation des tables cible soit défini sur Ne rien faire ou Tronquer. Dans ce cas, vous avez demandé à AWS DMS de ne pas effectuer de configuration sur les tables cibles, notamment de créer des index principaux et uniques. Si vous n'avez pas créé de clés primaires ou uniques sur les tables cibles, AWS DMS effectue une analyse complète de table pour chaque mise à jour. Cette approche peut avoir un impact considérable sur les performances.

Erreurs de violation de clé primaire lors du redémarrage d'une tâche

Cette erreur peut se produire lorsque des données restent dans la base de données cible depuis une tâche de migration précédente. Si l'option Mode de préparation des tables cible est définie sur Ne rien faire, AWS DMS ne réalise aucun préparation sur la table cible, y compris le nettoyage des données insérées à partir d'une tâche précédente.

Pour redémarrer votre tâche et éviter ces erreurs, supprimez les lignes insérées dans les tables cibles à partir de l'exécution précédente de la tâche.

Échec du chargement initial d'un schéma

Dans certains cas, le chargement initial de vos schémas peut échouer avec une erreur `Operation:getSchemaListDetails:errType=, status=0, errMessage=, errDetails=`.

Dans de tels cas, le compte d'utilisateur utilisé par AWS DMS pour se connecter au point de terminaison source ne dispose pas des autorisations nécessaires.

Échec des tâches avec une erreur inconnue

La cause des types d'erreur inconnus peut être variée. Toutefois, nous constatons souvent que le problème implique des ressources insuffisantes allouées à l'instance de réplication AWS DMS.

Pour garantir que votre instance dispose de suffisamment de ressources pour effectuer la migration, vérifiez l'utilisation que fait votre instance du CPU, de la mémoire, des fichiers d'échange et des IOPS. Pour plus d'informations concernant la supervision, consultez la page Métriques AWS Database Migration Service

Le redémarrage d'une tâche charge les tables dès le début

AWS DMS redémarre le chargement de table au début lorsque le chargement initial d'une table ne s'est pas terminé. Lorsqu'une tâche est redémarrée, AWS DMS recharge les tables depuis le début lorsque le chargement initial ne s'est pas terminé.

Le nombre de tables par tâche pose problème

Aucune limite ne s'applique au nombre de tables par tâche de réplication. Toutefois, en règle générale, nous recommandons de limiter le nombre de tables d'une tâche à moins de 60 000. L'utilisation des ressources peut souvent être un goulot d'étranglement lorsqu'une tâche unique utilise plus de 60 000 tables.

Les tâches échouent quand une clé primaire est créée sur une colonne LOB

En mode LOB COMPLET ou LOB LIMITÉ, AWS DMS ne prend pas en charge la réplication des clés primaires qui sont des types de données LOB.

DMS effectue initialement la migration d'une ligne avec une colonne LOB comme null, puis met à jour ultérieurement la colonne LOB. Ainsi, lorsque la clé primaire est créée sur une colonne LOB, l'insertion initiale échoue car la clé primaire ne peut pas être null. Comme solution de contournement, ajoutez une autre colonne en tant que clé primaire et supprimez la clé primaire de la colonne LOB.

Des doublons d'enregistrements apparaissent sur une table cible sans clé primaire

L'exécution d'une tâche de chargement complet + CDC peut créer des enregistrements en double sur les tables cibles sans clé primaire ni index unique. Pour éviter de dupliquer les enregistrements sur les tables cibles pendant les tâches de chargement complet et de CDC, assurez-vous que les tables cibles ont une clé primaire ou un index unique.

Échec des points de terminaison sources dans la plage IP réservée

Si une base de données source AWS DMS utilise une adresse IP comprise dans la plage IP réservée 192.168.0.0/24, le test de connexion des points de terminaison sources échoue. Les étapes suivantes fournissent une solution de contournement possible :

1. Recherchez une instance Amazon EC2 qui ne figure pas dans la plage réservée et qui peut communiquer avec la base de données source dans 192.168.0.0/24.
2. Installez un proxy socat et exécutez-le. Vous en trouverez un exemple ci-dessous.

```
yum install socat
```

```
socat -d -d -lmlocal2 tcp4-listen:database port,bind=0.0.0.0,reuseaddr,fork
 tcp4:source_database_ip_address:database_port
&
```

Utilisez l'adresse IP de l'instance Amazon EC2 et le port de base de données indiqué ci-dessus pour le point de terminaison AWS DMS. Assurez-vous que le point de terminaison dispose du groupe de sécurité qui permet à AWS DMS d'accéder au port de base de données. Notez que le proxy doit être actif pendant toute la durée d'exécution de votre tâche DMS. Selon le cas d'utilisation, vous devrez peut-être automatiser la configuration du proxy.

Les horodatages sont brouillés dans les requêtes Amazon Athena

Si les horodatages sont déformés dans les requêtes Athena, utilisez l'ModifyEndpointaction AWS Management Console ou pour définir la valeur de `parquetTimestampInMillisecond` votre point de terminaison Amazon S3 sur. `true` Pour plus d'informations, consultez S3Settings.

Résolution de problèmes avec Oracle

Vous allez en apprendre davantage sur la résolution des problèmes spécifiques à l'utilisation d'AWS DMS avec des bases de données Oracle.

Rubriques

- Extraire les données à partir de vues
- Migration des LOB à partir d'Oracle 12c
- Basculer entre Oracle LogMiner et Binary Reader
- Erreur : CDC Oracle arrêtée 122301 nombre maximal de nouvelles tentatives de CDC Oracle dépassé.
- Ajout automatique d'une journalisation supplémentaire à un point de terminaison source Oracle
- Les modifications de LOB ne sont pas capturées
- Erreur : ORA-12899 : valeur trop grande pour la colonne nom de la colonne
- Mauvaise interprétation du type de données NUMBER
- Enregistrements manquants pendant le chargement complet
- Erreur de table

- Erreur : impossible de récupérer les identifiants de destination des journaux Redo archivés par Oracle
- Évaluation des performances de lecture des journaux redo ou d'archivage Oracle

Extraire les données à partir de vues

Vous pouvez extraire une fois les données d'une vue ; vous ne pouvez pas les utiliser pour la réplication continue. Pour pouvoir extraire des données à partir de vues, vous devez ajouter le code suivant à la section Paramètres du point de terminaison de la page du point de terminaison source Oracle. Lorsque vous extrayez les données d'une vue, la vue est représentée sous la forme d'une table sur le schéma cible.

```
"ExposeViews": true
```

Migration des LOB à partir d'Oracle 12c

AWS DMSpeut utiliser deux méthodes pour capturer les modifications apportées à une base de données Oracle, Binary Reader et Oracle LogMiner. Par défaut, AWS DMS utilise Oracle LogMiner pour capturer les modifications. Toutefois, sur Oracle 12c, Oracle LogMiner ne prend pas en charge les colonnes LOB. Pour capturer les modifications apportées aux colonnes LOB dans Oracle 12c, utilisez Binary Reader.

Basculer entre Oracle LogMiner et Binary Reader

AWS DMSpeut utiliser deux méthodes pour capturer les modifications apportées à une base de données Oracle source, Binary Reader et Oracle LogMiner. Oracle LogMiner est la valeur par défaut. Pour utiliser Binary Reader pour capturer les modifications, procédez comme suit :

Pour utiliser Binary Reader pour capturer des modifications

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Endpoints (Points de terminaison).
3. Choisissez le point de terminaison source Oracle que vous souhaitez utiliser avec Binary Reader.
4. Sélectionnez Modifier.
5. Choisissez Avancé, puis ajoutez le code suivant pour Attributs de connexion supplémentaires.

```
useLogminerReader=N
```

6. Utilisez un outil de développement Oracle tel que SQL-Plus pour accorder le privilège supplémentaire suivant au compte d'utilisateur AWS DMS utilisé pour vous connecter au point de terminaison Oracle.

```
SELECT ON V_$TRANSPORTABLE_PLATFORM
```

Erreur : CDC Oracle arrêtée 122301 nombre maximal de nouvelles tentatives de CDC Oracle dépassé.

Cette erreur se produit lorsque les journaux d'archive Oracle nécessaires ont été supprimés de votre serveur avant qu'AWS DMS ait pu les utiliser pour capturer les modifications. Augmentez vos stratégies de conservation des journaux sur votre serveur de base de données. Pour une base de données Amazon RDS, exécutez la procédure suivante pour augmenter la durée de conservation des journaux. Par exemple, le code suivant augmente la durée de conservation des journaux sur une instance DB Amazon RDS à 24 heures.

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24);
```

Ajout automatique d'une journalisation supplémentaire à un point de terminaison source Oracle

Par défaut, la journalisation supplémentaire d'AWS DMS est désactivée. Pour activer automatiquement la journalisation supplémentaire pour un point de terminaison source Oracle, procédez comme suit.

Pour ajouter la journalisation supplémentaire à un point de terminaison source Oracle

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'[adresse https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/).

2. Choisissez Endpoints (Points de terminaison).
3. Choisissez le point de terminaison source Oracle auquel vous souhaitez ajouter la journalisation supplémentaire.
4. Sélectionnez Modifier.
5. Choisissez Avancé, puis ajoutez le code suivant dans la zone de texte Attributs de connexion supplémentaires :

```
addSupplementalLogging=Y
```

6. Sélectionnez Modifier.

Les modifications de LOB ne sont pas capturées

Actuellement, une table doit avoir une clé primaire pour qu'AWS DMS puisse capturer les modifications LOB. Si une table qui contient des données de type LOB ne possède pas de clé primaire, vous pouvez réaliser plusieurs actions pour capturer les modifications LOB :

- Ajouter une clé primaire à la table. Il peut suffire d'ajouter une colonne ID et de la renseigner avec une séquence à l'aide d'un déclencheur.
- Créez une vue matérialisée de la table qui inclut un ID généré par le système en tant que clé primaire et migrez la vue matérialisée plutôt que la table.
- Créer une copie de secours logique, ajouter une clé primaire à la table et effectuer la migration à partir de la copie de secours logique.

Erreur : ORA-12899 : valeur trop grande pour la colonne *nom de la colonne*

L'erreur « ORA-12899 : valeur trop grande pour la colonne *column-name* » est souvent causée par quelques problèmes.

L'un de ces problèmes est dû à une incompatibilité entre les jeux de caractères utilisés par les bases de données source et cible.

Dans un autre de ces problèmes, les paramètres NLS (National Language Support) diffèrent entre les deux bases de données. Cette erreur survient souvent lorsque le paramètre

NLS_LENGTH_SEMANTICS de base de données source est défini à la valeur CHAR et que le paramètre NLS_LENGTH_SEMANTICS de la base de données cible est défini à la valeur BYTE.

Mauvaise interprétation du type de données NUMBER

Le type de données Oracle NUMBER est converti en divers types de données AWS DMS, selon la précision et l'échelle de NUMBER. Ces conversions sont documentées ici Types de données sources pour Oracle. La façon dont le type NUMBER est converti peut également être affectée par l'utilisation de paramètres de point de terminaison pour le point de terminaison Oracle source. Ces paramètres de point de terminaison sont documentés dans Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS.

Enregistrements manquants pendant le chargement complet

Lors d'un chargement complet, AWS DMS recherche les transactions ouvertes au niveau de la base de données et attend que la transaction soit validée. Par exemple, en fonction du paramètre de tâche `TransactionConsistencyTimeout=600`, AWS DMS attend 10 minutes même si la transaction ouverte porte sur une table non incluse dans le mappage de tables. Mais si la transaction ouverte porte sur une table incluse dans le mappage de tables et que la transaction n'est pas validée à temps, il en résulte des enregistrements manquants dans la table cible.

Vous pouvez modifier le paramètre de tâche `TransactionConsistencyTimeout` et augmenter le temps d'attente si vous savez que la validation des transactions ouvertes prendra plus de temps.

Notez également que la valeur par défaut du paramètre de tâche `FailOnTransactionConsistencyBreached` est `false`. Cela signifie qu'AWS DMS continue d'appliquer d'autres transactions mais que des transactions ouvertes sont manquées. Si vous souhaitez que la tâche échoue quand des transactions ouvertes ne sont pas fermées à temps, vous pouvez définir `FailOnTransactionConsistencyBreached` sur `true`.

Erreur de table

`Table Error` apparaît dans les statistiques de table lors de la réplication si une clause `WHERE` ne fait pas référence à une colonne de clé primaire et qu'une journalisation supplémentaire n'est pas utilisée pour toutes les colonnes.

Pour résoudre ce problème, activez la journalisation supplémentaire pour toutes les colonnes de la table référencée. Pour plus d'informations, consultez Configurez une journalisation supplémentaire.

Erreur : impossible de récupérer les identifiants de destination des journaux Redo archivés par Oracle

Cette erreur se produit quand aucun journal d'archive n'est généré pour votre source Oracle ou que V$ARCHIVED_LOG est vide. Vous pouvez résoudre cette erreur en échangeant les journaux manuellement.

Pour une base de données Amazon RDS, exécutez la procédure suivante pour échanger les fichiers journaux. La procédure `switch_logfile` ne comporte aucun paramètre.

```
exec rdsadmin.rdsadmin_util.switch_logfile;
```

Pour une base de données source Oracle autogérée, utilisez la commande suivante pour forcer un échange de journaux.

```
ALTER SYSTEM SWITCH LOGFILE ;
```

Évaluation des performances de lecture des journaux redo ou d'archivage Oracle

Si vous rencontrez des problèmes de performances avec votre source Oracle, vous pouvez évaluer les performances de lecture de vos journaux redo ou d'archivage Oracle afin de trouver des moyens d'améliorer les performances. Pour tester les performances de lecture des journaux redo ou d'archivage, utilisez l'Amazon Machine Image (AMI) de diagnostic AWS DMS.

Vous pouvez utiliser l'AMI de diagnostic AWS DMS pour effectuer les opérations suivantes :

- Utiliser la méthode bFile pour évaluer les performances des fichiers journaux redo.
- Utilisez LogMiner cette méthode pour évaluer les performances du fichier de journalisation.
- Utiliser la méthode PL/SQL (`dbms_lob.read`) pour évaluer les performances des fichiers journaux redo.
- Utiliser une exécution monothread pour évaluer les performances de lecture sur un fichier ASM.
- Utiliser une exécution multithread pour évaluer les performances de lecture sur un fichier ASM.
- Utiliser la fonction directe Windows Readfile() ou Linux Pread64 pour évaluer le fichier journal redo.

Vous pouvez ensuite prendre des mesures correctives en fonction des résultats.

Pour tester les performances de lecture sur un fichier journal redo ou d'archivage Oracle

1. Créez une instance Amazon EC2 d'AMI de diagnostic AWS DMS et connectez-vous à celle-ci.

   Pour plus d'informations, consultez Utilisation de l'AMI de diagnostic AWS DMS.

2. Exécutez la commande awsreplperf.

   ```
   $ awsreplperf
   ```

   Cette commande affiche les options de l'utilitaire de performances de lecture Oracle d'AWS DMS.

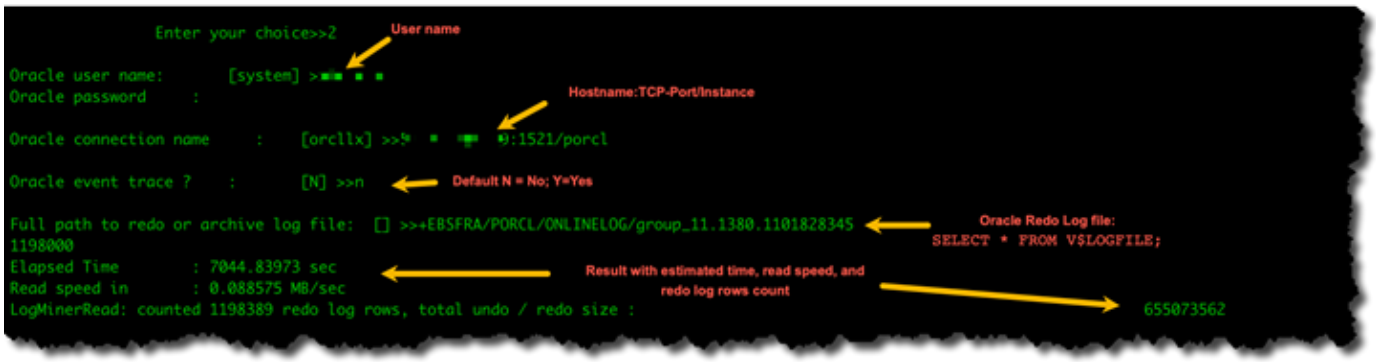
   ```
   0. Quit
   1. Read using Bfile
   2. Read using LogMiner
   3. Read file PL/SQL (dms_lob.read)
   4. Read ASMFile Single Thread
   5. Read ASMFile Multi Thread
   6. Readfile() function
   ```

3. Sélectionnez une option dans la liste.

4. Entrez les informations de journal d'archivage et de connexion de base de données suivantes.

   ```
   Oracle user name [system]:
   Oracle password:

   Oracle connection name [orcllx]:
   Connection format hostname:port/instance

   Oracle event trace? [N]:
   Default N = No or Y = Yes

   Path to redo or archive log file []:
   ```

5. Examinez la sortie affichée pour obtenir des informations pertinentes sur les performances de lecture. Par exemple, ce qui suit montre le résultat qui peut résulter de la sélection de l'option numéro 2, Lire en utilisant LogMiner.



6. Pour quitter l'utilitaire, entrez 0 (zéro).

Étapes suivantes

- Lorsque les résultats indiquent que la vitesse de lecture est inférieure à un seuil acceptable, exécutez le script d'assistance au diagnostic Oracle sur le point de terminaison et passez en revue les sections Temps d'attente, Charger le profil et Profil d'E/S. Ajustez ensuite toute configuration anormale susceptible d'améliorer les performances de lecture. Par exemple, si vos fichiers journaux redo ont une capacité maximale de 2 Go, essayez d'augmenter LOG_BUFFER à 200 Mo pour améliorer les performances.
- Passez en revue les bonnes pratiques AWS DMS pour vous assurer que votre instance, votre tâche et vos points de terminaison de réplication DMS sont configurés de manière optimale.

Résolution des problèmes liés à MySQL

Vous allez en apprendre davantage sur la résolution des problèmes spécifiques à l'utilisation d'AWS DMS avec des bases de données MySQL.

Rubriques

- Échec de la tâche CDC pour le point de terminaison d'instance DB Amazon RDS car la journalisation binaire est désactivée
- Les connexions à une instance MySQL cible sont déconnectées durant une tâche
- Ajout de la validation automatique à un point de terminaison compatible MySQL
- Désactiver les clés étrangères sur un point de terminaison cible compatible MySQL
- Caractères remplacés par un point d'interrogation
- Entrées de journal « Événement incorrect »

- Capture de données modifiées avec MySQL 5.5
- Augmentation de la durée de conservation des journaux binaires pour les instances de base de données Amazon RDS
- Message du journal : quelques modifications de la base de données source n'ont eu aucun impact lorsqu'elles ont été appliquées à la base de données cible.
- Erreur : Identificateur trop long
- Erreur : un jeu de caractères non pris en charge entraîne l'échec de la conversion des données de champ
- Erreur : page de codes 1252 à UTF8 [120112] Échec de la conversion des données d'un champ
- Non-migration des index, des clés étrangères ou des mises à jour ou suppressions en cascade

Échec de la tâche CDC pour le point de terminaison d'instance DB Amazon RDS car la journalisation binaire est désactivée

Ce problème se produit avec les instances de base de données Amazon RDS car les sauvegardes automatiques sont désactivées. Activez les sauvegardes automatiques en définissant la période de conservation des sauvegardes avec une valeur différente de zéro.

Les connexions à une instance MySQL cible sont déconnectées durant une tâche

Si une tâche contenant des objets LOB est déconnectée d'une cible MySQL, le journal des tâches peut présenter le type d'erreurs suivant.

```
[TARGET_LOAD ]E: RetCode: SQL_ERROR SqlState: 08S01 NativeError:
2013 Message: [MySQL][ODBC 5.3(w) Driver][mysqld-5.7.16-log]Lost connection
to MySQL server during query [122502] ODBC general error.
```

```
 [TARGET_LOAD ]E: RetCode: SQL_ERROR SqlState: HY000 NativeError:
2006 Message: [MySQL][ODBC 5.3(w) Driver]MySQL server has gone away
[122502] ODBC general error.
```

Dans ce cas, il se peut que vous deviez ajuster certains paramètres de tâche.

Pour résoudre le problème de déconnexion d'une tâche à une cible MySQL, procédez comme suit :

- Vérifiez que vos variables de base de données `max_allowed_packet` sont assez grandes pour contenir la plus grande taille de LOB.
- Vérifiez que les variables suivantes sont définies pour présenter une valeur de délai d'expiration élevée. Nous vous recommandons d'utiliser une valeur d'au moins 5 minutes pour chacune de ces variables.
  - `net_read_timeout`
  - `net_write_timeout`
  - `wait_timeout`

Pour plus d'informations sur la définition des variables système MySQL, consultez Variables système du serveur (langue française non garantie) dans la documentation MySQL.

Ajout de la validation automatique à un point de terminaison compatible MySQL

Pour ajouter la validation automatique à un point de terminaison cible compatible MySQL

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Endpoints (Points de terminaison).
3. Choisissez le point de terminaison cible compatible MySQL auquel vous souhaitez ajouter la validation automatique.
4. Sélectionnez Modifier.
5. Choisissez Avancé, puis ajoutez le code suivant dans la zone de texte Attributs de connexion supplémentaires :

```
Initstmt= SET AUTOCOMMIT=1
```

6. Sélectionnez Modifier.

Désactiver les clés étrangères sur un point de terminaison cible compatible MySQL

Vous pouvez désactiver les contrôles de clé étrangère sur MySQL en ajoutant les éléments suivants dans Attributs de connexion supplémentaires, dans la section Avancé du point de terminaison cible MySQL, Amazon Aurora MySQL-Compatible Edition ou MariaDB.

Pour désactiver les clés étrangères sur un point de terminaison cible compatible MySQL

1. Connectez-vous à la AWS DMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/dms/v2/.
2. Choisissez Endpoints (Points de terminaison).
3. Choisissez le point de terminaison cible MySQL, Aurora MySQL ou MariaDB pour lequel vous voulez désactiver les clés étrangères.
4. Sélectionnez Modifier.
5. Choisissez Avancé, puis ajoutez le code suivant dans la zone de texte Attributs de connexion supplémentaires :

```
Initstmt=SET FOREIGN_KEY_CHECKS=0
```

6. Sélectionnez Modifier.

Caractères remplacés par un point d'interrogation

Ce problème se produit suivant lorsque les caractères de point de terminaison source ont été codés par un jeu de caractères non pris en charge par AWS DMS.

Entrées de journal « Événement incorrect »

Les entrées « Événement incorrect » dans les journaux de migration indiquent généralement qu'une opération de langage de manipulation de données (DDL) non prise en charge a été tentée sur le point de terminaison de la base de données source. Les opérations DDL non prises en charge déclenchent un événement que l'instance de réplication ne peut pas ignorer, si bien qu'un événement incorrect est journalisé.

Pour résoudre ce problème, redémarrez la tâche depuis le début. Cela recharge les tables et lance la capture des modifications une fois que l'opération DDL non prise en charge a été émise.

Capture de données modifiées avec MySQL 5.5

La capture des données de modification (CDC) AWS DMS pour les bases de données compatibles Amazon RDS MySQL nécessite la journalisation binaire basée sur les lignes de l'image complète, qui n'est pas prise en charge dans MySQL versions 5.5 et antérieures. Pour pouvoir utiliser la CDC AWS DMS, vous devez mettre à niveau votre instance de base de données Amazon RDS à MySQL version 5.6.

Augmentation de la durée de conservation des journaux binaires pour les instances de base de données Amazon RDS

AWS DMS nécessite la conservation des fichiers journaux binaires pour la capture de données modifiées. Pour augmenter la durée de conservation des journaux sur une instance DB Amazon RDS, utilisez la procédure suivante. L'exemple suivant permet d'augmenter la durée de conservation des journaux binaires à 24 heures.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Message du journal : quelques modifications de la base de données source n'ont eu aucun impact lorsqu'elles ont été appliquées à la base de données cible.

Lorsque AWS DMS met à jour la valeur d'une colonne de base de données MySQL à sa valeur existante, un message `zero rows affected`est renvoyé à partir de MySQL. Ce comportement est différent des autres moteurs de base de données tels qu'Oracle et SQL Server. Ces moteurs mettent à jour une ligne, même si la valeur de remplacement est identique à la valeur actuelle.

Erreur : Identificateur trop long

L'erreur suivante se produit lorsqu'un identificateur est trop long :

```
TARGET_LOAD E: RetCode: SQL_ERROR SqlState: HY000 NativeError:
```

```
1059 Message: MySQLhttp://ODBC 5.3(w) Driverhttp://mysqld-5.6.10Identifier
name 'name' is too long 122502 ODBC general error. (ar_odbc_stmt.c:4054)
```

Dans certains cas, vous configurez AWS DMS pour créer les tables et les clés primaires dans la base de données cible. Dans de tels cas, DMS n'utilise actuellement pas pour les clés primaires les noms utilisés dans la base de données source. Au lieu de cela, DMS crée le nom de clé primaire en fonction du nom de la table. Lorsque le nom de la table est long, l'identificateur généré automatiquement peut être plus long que la limite autorisée pour MySQL.

Pour résoudre ce problème, l'approche actuelle consiste à précréer les tables et les clés primaires dans la base de données cible. Utilisez ensuite une tâche dont le paramètre de tâche Mode de préparation des tables cible est défini sur Ne rien faire ou Tronquer pour remplir les tables cibles.

Erreur : un jeu de caractères non pris en charge entraîne l'échec de la conversion des données de champ

L'erreur suivante se produit lorsqu'un jeu de caractères non pris en charge entraîne l'échec de la conversion des données de champ :

```
"[SOURCE_CAPTURE ]E: Column 'column-name' uses an unsupported character set [120112]
A field data conversion failed. (mysql_endpoint_capture.c:2154)
```

Vérifiez les paramètres de la base de données liés aux connexions. La commande suivante peut être utilisée pour définir ces paramètres.

```
SHOW VARIABLES LIKE '%char%';
```

Erreur : page de codes 1252 à UTF8 [120112] Échec de la conversion des données d'un champ

L'erreur suivante peut se produire pendant une migration si des caractères autres que page de codes 1252 se trouvent dans la base de données MySQL source.

```
[SOURCE_CAPTURE ]E: Error converting column 'column_xyz' in table
'table_xyz with codepage 1252 to UTF8 [120112] A field data conversion failed.
(mysql_endpoint_capture.c:2248)
```

Pour contourner ce problème, vous pouvez utiliser l'attribut de connexion supplémentaire `CharsetMapping` avec votre point de terminaison MySQL source pour spécifier la correspondance des jeux de caractères. Vous devrez peut-être redémarrer la tâche de migration AWS DMS depuis le début si vous ajoutez ce paramètre de point de terminaison.

Par exemple, le paramètre de point de terminaison suivant peut être utilisé pour un point de terminaison source MySQL dont le jeu de caractères source est `Utf8` ou `latin1`. 65001 est l'identificateur de la page de code UTF8.

```
CharsetMapping=utf8,65001
CharsetMapping=latin1,65001
```

Non-migration des index, des clés étrangères ou des mises à jour ou suppressions en cascade

AWS DMS ne prend pas en charge la migration d'objets secondaires tels que les index et les clés étrangères. Pour répliquer les modifications apportées aux tables enfants à la suite d'une opération de mise à jour ou de suppression en cascade, la contrainte de clé étrangère de déclenchement doit être active sur la table cible. Pour contourner cette limitation, créez la clé étrangère manuellement sur la table cible. Créez ensuite une tâche unique pour le chargement complet et la CDC, ou deux tâches distinctes pour le chargement complet et la CDC, comme décrit ci-après :

Création d'une tâche unique prenant en charge le chargement complet et la CDC

Cette procédure décrit comment migrer des clés étrangères et des index à l'aide d'une tâche unique de chargement complet et de CDC.

Création d'une tâche de chargement complet et de CDC

1. Créez manuellement les tables avec des clés étrangères et des index sur la cible pour qu'elles correspondent aux tables sources.
2. Ajoutez l'attribut ECA suivant au point de terminaison AWS DMS cible :

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

3. Créez la tâche AWS DMS avec `TargetTablePrepMode` défini sur `DO_NOTHING`.
4. Définissez le paramètre `Stop task after full load completes` sur `StopTaskCachedChangesApplied`.
5. Lancez la tâche. AWS DMS arrête automatiquement la tâche une fois le chargement complet terminé et applique les modifications mises en cache.
6. Supprimez l'attribut ECA `SET FOREIGN_KEY_CHECKS` que vous avez ajouté précédemment.
7. Reprenez la tâche. La tâche entre dans la phase CDC et applique les modifications continues de la base de données source à la cible.

Création séparée de tâches de chargement complet et de CDC

Ces procédures décrivent comment migrer les clés étrangères et les index à l'aide de tâches distinctes de chargement complet et de CDC.

Création d'une tâche de chargement complet

1. Créez manuellement les tables avec des clés étrangères et des index sur la cible pour qu'elles correspondent aux tables sources.
2. Ajoutez l'attribut ECA suivant au point de terminaison AWS DMS cible :

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

3. Créez la tâche AWS DMS avec le paramètre `TargetTablePrepMode` défini sur `DO_NOTHING` et `EnableValidation` défini sur `FALSE`.
4. Lancez la tâche. AWS DMS arrête automatiquement la tâche une fois le chargement complet terminé et applique les modifications mises en cache.
5. Une fois la tâche terminée, notez l'heure UTC de début de la tâche de chargement complet, ou le nom et la position du fichier journal binaire, pour démarrer la tâche de CDC uniquement. Reportez-vous aux journaux pour obtenir l'horodatage (UTC) à partir de l'heure de début du chargement complet initial.

Création d'une tâche de CDC uniquement

1. Supprimez l'attribut ECA `SET FOREIGN_KEY_CHECKS` que vous avez défini précédemment.

2. Créez la tâche de CDC uniquement avec la position de départ définie sur l'heure de début du chargement complet, notée à l'étape précédente. Vous pouvez également utiliser la position du journal binaire enregistrée à l'étape précédente. Définissez le paramètre `TargetTablePrepMode` sur `DO_NOTHING`. Activez la validation des données en définissant le paramètre `EnableValidation` sur `TRUE`, si nécessaire.
3. Démarrez la tâche de CDC uniquement et surveillez les journaux pour détecter les erreurs.

> **Note**
>
> Cette solution de contournement s'applique uniquement à une migration de MySQL vers MySQL. Vous ne pouvez pas utiliser cette méthode avec la fonctionnalité d'application par lots, car l'application par lots nécessite que les tables cibles ne possèdent pas de clés étrangères actives.

Résolution des problèmes liés à PostgreSQL

Vous allez en apprendre davantage sur la résolution des problèmes spécifiques à l'utilisation d'AWS DMS avec des bases de données PostgreSQL.

Rubriques

- Types de données JSON tronqués
- Migration incorrecte de colonnes d'un type de données défini par l'utilisateur
- Erreur : Aucun schéma sélectionné dans lequel effectuer la création
- Les suppressions et les mises à jour dans une table ne sont pas répliquées via la CDC
- Les instructions de troncation ne sont pas propagées
- Empêcher PostgreSQL de capturer la DDL
- Sélectionner le schéma où sont créés les objets de base de données pour la capture de la DDL
- Tables Oracle manquantes après la migration vers PostgreSQL
- ReplicationSlotDiskUsage augmente et restart_lsn cesse d'avancer pendant les transactions longues, telles que les charges de travail ETL
- Une tâche utilisant une vue comme source ne contient aucune ligne copiée

Types de données JSON tronqués

AWS DMS traite le type de données JSON dans PostgreSQL sous la forme d'une colonne de type de données LOB. Cela signifie que la limite de taille des objets LOB, lorsque vous utilisez le mode LOB limité, s'applique aux données JSON.

Supposons, par exemple, que le mode LOB limité soit défini sur 4 096 Ko. Dans ce cas, toutes les données JSON d'une taille supérieure à 4 096 Ko sont tronquées à la limite de 4 096 Ko et échouent au test de validation dans PostgreSQL.

Les informations de journalisation suivantes montrent des données JSON qui ont été tronquées en raison du mode LOB limité et qui n'ont pas pu être validées.

```
03:00:49
2017-09-19T03:00:49 [TARGET_APPLY ]E: Failed to execute statement:
  'UPDATE "public"."delivery_options_quotes" SET "id"=? , "enabled"=? ,
  "new_cart_id"=? , "order_id"=? , "user_id"=? , "zone_id"=? , "quotes"=? ,
  "start_at"=? , "end_at"=? , "last_quoted_at"=? , "created_at"=? ,
  "updated_at"=? WHERE "id"=? ' [1022502] (ar_odbc_stmt
2017-09-19T03:00:49 [TARGET_APPLY ]E: Failed to execute statement:
  'UPDATE "public"."delivery_options_quotes" SET "id"=? , "enabled"=? ,
  "new_cart_id"=? , "order_id"=? , "user_id"=? , "zone_id"=? , "quotes"=? ,
  "start_at"=? , "end_at"=? , "last_quoted_at"=? , "created_at"=? ,
  "updated_at"=? WHERE "id"=? ' [1022502] (ar_odbc_stmt.c:2415)
#
03:00:49
2017-09-19T03:00:49 [TARGET_APPLY ]E: RetCode: SQL_ERROR SqlState:
  22P02 NativeError: 1 Message: ERROR: invalid input syntax for type json;,
  Error while executing the query [1022502] (ar_odbc_stmt.c:2421)
2017-09-19T03:00:49 [TARGET_APPLY ]E: RetCode: SQL_ERROR SqlState:
  22P02 NativeError: 1 Message: ERROR: invalid input syntax for type json;,
  Error while executing the query [1022502] (ar_odbc_stmt.c:2421)
```

Migration incorrecte de colonnes d'un type de données défini par l'utilisateur

Lors de la réplication à partir d'une source PostgreSQL, AWS DMS crée la table cible avec les mêmes types de données pour toutes les colonnes, sauf les colonnes contenant des types de données définis par l'utilisateur. Dans ces cas, le type de données est créé en tant que « character varying » dans la cible.

Erreur : Aucun schéma sélectionné dans lequel effectuer la création

Dans certains cas, le message d'erreur « SQL_ERROR SqlState : 3F000:7 Message : ERREUR NativeError : aucun schéma n'a été sélectionné pour être créé » peut s'afficher.

Cette erreur peut se produire lorsque le mappage de table JSON contient une valeur générique pour le schéma mais que la base de données source ne prend pas en charge cette valeur.

Les suppressions et les mises à jour dans une table ne sont pas répliquées via la CDC

Les opérations de suppression et de mise à jour pendant la capture des données de modification (CDC) sont ignorées si la table source ne possède pas de clé primaire. AWS DMS prend en charge la capture des données de modification (CDC) pour les tables PostgreSQL avec des clés primaires.

Si une table n'a pas de clé primaire, les journaux d'écriture anticipée (WAL) n'incluent pas d'image antérieure de la ligne de base de données. Dans ce cas, AWS DMS ne peut pas mettre à jour la table. Pour que les opérations de suppression soient répliquées, créez une clé primaire sur la table source.

Les instructions de troncation ne sont pas propagées

Lorsque vous utilisez la capture des données de modification (CDC), les opérations TRUNCATE ne sont pas prises en charge par AWS DMS.

Empêcher PostgreSQL de capturer la DDL

Vous pouvez empêcher un point de terminaison cible PostgreSQL de capturer des instructions DDL en ajoutant l'instruction Paramètres du point de terminaison suivante.

```
"CaptureDDLs": "N"
```

Sélectionner le schéma où sont créés les objets de base de données pour la capture de la DDL

Vous pouvez contrôler dans quel schéma les objets de base de données liés à la capture de la DDL sont créés. Ajoutez l'instruction Paramètres du point de terminaison suivante. Le paramètre Paramètres du point de terminaison est disponible dans l'onglet du point de terminaison source.

```
"DdlArtifactsSchema: "xyzddlschema"
```

Tables Oracle manquantes après la migration vers PostgreSQL

Dans ce cas, vos tables et données restent généralement accessibles.

Par défaut, Oracle met en majuscules les noms de table, alors que PostgreSQL les met en minuscules par défaut. Lorsque vous effectuez une migration d'Oracle vers PostgreSQL, nous vous conseillons de fournir certaines règles de transformation dans la section de mappage de table de votre tâche. Il s'agit de règles de transformation permettant de convertir la casse des noms de table.

Si vous avez migré des tables sans utiliser de règles de transformation pour convertir la casse des noms de table, placez les noms des tables entre guillemets lorsque vous les référencez.

ReplicationSlotDiskUsage augmente et restart_lsn cesse d'avancer pendant les transactions longues, telles que les charges de travail ETL

Lorsque la réplication logique est activée, le nombre maximal de modifications conservées en mémoire par transaction est de 4 Mo. Ensuite, les modifications sont déversées sur le disque. En conséquence, `ReplicationSlotDiskUsage` augmente et `restart_lsn` n'avance pas tant que la transaction n'est pas terminée ou abandonnée et que la restauration n'est pas terminée. Comme il s'agit d'une transaction longue, sa restauration peut prendre beaucoup de temps.

Vous devez donc éviter les transactions de longue durée lorsque la réplication logique est activée. Essayez plutôt de diviser la transaction en plusieurs transactions plus petites.

Une tâche utilisant une vue comme source ne contient aucune ligne copiée

Pour migrer une vue, définissez `table-type` sur `all` ou `view`. Pour plus d'informations, consultez Spécification des règles de sélection de table et de transformation à partir de la console.

Les sources qui prennent en charge les vues incluent les suivantes.

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2 LUW
- SAP Adaptive Server Enterprise (ASE)

Résolution des problèmes liés à Microsoft SQL Server

Vous allez en apprendre davantage sur la résolution des problèmes spécifiques à l'utilisation d'AWS DMS avec des bases de données Microsoft SQL Server.

Rubriques

- Erreurs de capture des modifications pour une base de données SQL Server
- Colonnes d'identité manquantes
- Erreur : SQL Server ne prend pas en charge les publications
- Les modifications ne s'affichent pas dans votre cible
- Table non uniforme mappée entre les partitions

Erreurs de capture des modifications pour une base de données SQL Server

Les erreurs pendant la capture des données de modification (CDC) indiquent souvent qu'une des conditions préalables n'était pas remplie. Par exemple, l'une des conditions préalables les plus fréquemment négligées est une sauvegarde complète de la base de données. Le journal des tâches indique cette omission par l'erreur suivante :

```
SOURCE_CAPTURE E: No FULL database backup found (under the 'FULL' recovery model).
To enable all changes to be captured, you must perform a full database backup.
120438 Changes may be missed. (sqlserver_log_queries.c:2623)
```

Passez en revue les conditions préalables répertoriées pour l'utilisation de SQL Server en tant que source dans Utilisation d'une base de données Microsoft SQL Server comme source pour AWS DMS.

Colonnes d'identité manquantes

AWS DMS ne prend pas en charge les colonnes d'identité lorsque vous créez un schéma cible. Vous devez les ajouter une fois le chargement initial terminé.

Erreur : SQL Server ne prend pas en charge les publications

L'erreur suivante est généré lorsque vous utilisez SQL Server Express comme point de terminaison source :

```
RetCode: SQL_ERROR SqlState: HY000 NativeError: 21106
Message: This edition of SQL Server does not support publications.
```

AWS DMS ne prend pas en charge actuellement SQL Server Express en tant que source ou cible.

Les modifications ne s'affichent pas dans votre cible

AWS DMS nécessite qu'une base de données SQL Server source soit en mode de récupération de données « FULL » ou « BULK LOGGED » afin de capturer systématiquement les modifications. Le modèle « SIMPLE » n'est pas pris en charge.

Le modèle de récupération SIMPLE enregistre les informations minimales nécessaires pour permettre aux utilisateurs de récupérer leur base de données. Toutes les entrées de journal inactives sont automatiquement tronquées lorsqu'un point de contrôle se produit.

Toutes les opérations sont toujours journalisées. Toutefois, dès qu'un point de contrôle se produit, le journal est automatiquement tronqué. Cette troncation signifie que le journal peut être réutilisé et que les anciennes entrées du journal peuvent être remplacées. Lorsque les entrées du journal sont remplacées, les modifications ne peuvent pas être capturées. Ce problème explique pourquoi AWS DMS ne prend pas en charge le modèle de récupération de données SIMPLE. Pour en savoir plus sur les autres conditions préalables requises pour l'utilisation de SQL Server en tant que source, consultez Utilisation d'une base de données Microsoft SQL Server comme source pour AWS DMS.

Table non uniforme mappée entre les partitions

Pendant la capture des données de modification (CDC), la migration d'une table dotée d'une structure spécialisée est suspendue quand AWS DMS ne peut pas effectuer correctement le processus CDC sur la table. Des messages similaires aux suivants sont émis :

```
[SOURCE_CAPTURE ]W: Table is not uniformly mapped across partitions. Therefore - it is
 excluded from CDC (sqlserver_log_metadata.c:1415)
[SOURCE_CAPTURE ]I: Table has been mapped and registered for CDC.
 (sqlserver_log_metadata.c:835)
```

Lors de l'exécution du processus CDC sur des tables SQL Server, AWS DMS analyse les enregistrements tlog SQL Server. Sur chaque enregistrement tlog, AWS DMS analyse les valeurs hexadécimales contenant des données pour les colonnes qui ont été insérées, mises à jour ou supprimées lors d'une modification.

Pour analyser l'enregistrement hexadécimal, AWS DMS lit les métadonnées de la table à partir des tables système SQL Server. Ces tables système identifient ce que sont les colonnes de table spécialement structurées et révèlent certaines de leurs propriétés internes, telles que « xoffset » et « null bit position ».

AWS DMS s'attend à ce que les métadonnées soient identiques pour toutes les partitions brutes de la table. Toutefois, dans certains cas, les tables spécialement structurées n'ont pas les mêmes métadonnées sur toutes leurs partitions. Dans ces cas, AWS DMS peut suspendre le processus CDC sur cette table pour éviter d'analyser les modifications de manière incorrecte et de fournir à la cible des données incorrectes. Les solutions de contournement incluent les suivantes :

- Si la table a un index cluster, effectuez une reconstruction de l'index.
- Si la table n'a pas d'index organisé en cluster, ajoutez un index organisé en cluster à la table (vous pouvez le supprimer ultérieurement si vous le souhaitez).

Résolution des problèmes liés à Amazon Redshift

Vous allez en apprendre davantage sur la résolution des problèmes spécifiques à l'utilisation d'AWS DMS avec des bases de données Amazon Redshift.

Rubriques

- Chargement dans un cluster Amazon Redshift dans une autre région AWS
- Erreur : la relation « awsdms_apply_exceptions » existe déjà
- Erreurs avec les tables dont le nom commence par « awsdms_changes »
- Présence de tables dans les clusters avec des noms du type dms.awsdms_changes000000000XXXX
- Autorisations requises pour utiliser Amazon Redshift

Chargement dans un cluster Amazon Redshift dans une autre région AWS

Vous ne pouvez pas charger dans un cluster Amazon Redshift dans une région AWS autre que celle de l'instance de réplication AWS DMS. DMS exige que l'instance de réplication et le cluster Amazon Redshift figurent dans la même région.

Erreur : la relation « awsdms_apply_exceptions » existe déjà

L'erreur « La relation « awsdms_apply_exceptions » existe déjà » se produit souvent lorsqu'un point de terminaison Redshift est spécifié en tant que point de terminaison PostgreSQL. Pour résoudre ce problème, modifiez le point de terminaison et le Moteur cible en « redshift ».

Erreurs avec les tables dont le nom commence par « awsdms_changes »

Des messages d'erreur de table avec des noms commençant par « awsdms_changes » peuvent apparaître quand deux tâches qui tentent de charger des données dans le même cluster Amazon Redshift s'exécutent simultanément. En raison de la façon dont les tables temporaires sont nommés, des tâches simultanées peuvent entrer en conflit lors de la mise à jour d'une même table.

Présence de tables dans les clusters avec des noms du type dms.awsdms_changes000000000XXXX

AWS DMS crée des tables temporaires lorsque les données sont chargées à partir de fichiers stockés dans Amazon S3. Les noms de ces tables temporaires contiennent tous le préfixe `dms.awsdms_changes`. Ces tables sont obligatoires pour que AWS DMS puisse stocker des données lorsqu'elles sont chargées pour la première fois et avant qu'elles soient placées dans la table cible finale.

Autorisations requises pour utiliser Amazon Redshift

Pour utiliser AWS DMS avec Amazon Redshift, le compte d'utilisateur que vous utilisez pour accéder à Amazon Redshift doit disposer des autorisations suivantes :

- CRUD (choisir, insérer, mettre à jour, supprimer)
- Chargement en bloc
- Création, modification, suppression (si requis par la définition de la tâche)

Pour voir les conditions préalables requises pour l'utilisation d'Amazon Redshift en tant que cible, consultez Utilisation d'une base de données Amazon Redshift en tant que cible pour AWS Database Migration Service.

Résolution des problèmes liés à Amazon Aurora MySQL

Vous allez en apprendre davantage sur la résolution des problèmes spécifiques à l'utilisation d'AWS DMS avec des bases de données Amazon Aurora MySQL.

Rubriques

- Erreur : Champs CHARACTER SET UTF8 se terminant par une « , » entourée par des lignes « " » se terminant par « \n »

Erreur : Champs CHARACTER SET UTF8 se terminant par une « , » entourée par des lignes « " » se terminant par « \n »

Si vous utilisez Amazon Aurora MySQL en tant que cible, il est possible que vous voyiez une erreur similaire à la suivante dans les journaux. Ce type d'erreur indique généralement que le paramètre SQL_MODE contient des caractères ANSI_QUOTES. Si le paramètre SQL_MODE contient des caractères ANSI_QUOTES, les guillemets doubles sont traités comme des apostrophes et peuvent générer des problèmes lorsque vous exécutez une tâche.

Pour corriger cette erreur, supprimez les caractères ANSI_QUOTES du paramètre SQL_MODE.

```
2016-11-02T14:23:48 [TARGET_LOAD ]E: Load data sql statement. load data local infile
"/rdsdbdata/data/tasks/7XO4FJHCVON7TYTLQ6RX3CQHDU/data_files/4/LOAD000001DF.csv" into
 table
```

```
`VOSPUSER`.`SANDBOX_SRC_FILE` CHARACTER SET UTF8 fields terminated by ','
enclosed by '"' lines terminated by '\n'( `SANDBOX_SRC_FILE_ID`,`SANDBOX_ID`,
`FILENAME`,`LOCAL_PATH`,`LINES_OF_CODE`,`INSERT_TS`,`MODIFIED_TS`,`MODIFIED_BY`,
`RECORD_VER`,`REF_GUID`,`PLATFORM_GENERATED`,`ANALYSIS_TYPE`,`SANITIZED`,`DYN_TYPE`,
`CRAWL_STATUS`,`ORIG_EXEC_UNIT_VER_ID` ) ; (provider_syntax_manager.c:2561)
```

Résolution des problèmes liés à SAP ASE

Vous allez en apprendre davantage sur la résolution des problèmes spécifiques à l'utilisation d'AWS DMS avec des bases de données SAP ASE.

Erreur : les colonnes LOB ont des valeurs NULL lorsque la source possède un index unique composite avec des valeurs NULL

Lorsque vous utilisez SAP ASE en tant que source avec des tables configurées avec un index unique composite qui autorise les valeurs NULL, les valeurs LOB risquent de ne pas migrer pendant la réplication continue. Ce comportement est généralement dû au fait que la valeur ANSI_NULL est définie sur 1 par défaut sur le client de l'instance de réplication DMS.

Pour vous assurer que les champs LOB soient correctement migrés, incluez le paramètre de point de terminaison `'AnsiNull=0'` dans le point de terminaison source AWS DMS de la tâche.

Résolution des problèmes liés à IBM Db2

Vous allez en apprendre davantage sur la résolution des problèmes spécifiques à l'utilisation d'AWS DMS avec des bases de données IBM Db2.

Erreur : Resume from timestamp is not supported Task

Pour la réplication continue (CDC), si vous prévoyez de démarrer la réplication à partir d'un horodatage spécifique, définissez l'attribut de connexion `StartFromContext` sur l'horodatage requis. Pour plus d'informations, consultez Paramètres de point de terminaison lors de l'utilisation de Db2 LUW. Définir `StartFromContext` sur l'horodatage requis permet d'éviter le problème suivant :

```
Last Error Resume from timestamp is not supported Task error notification received
 from
subtask 0, thread 0 [reptask/replicationtask.c:2822] [1020455] 'Start from timestamp'
 was blocked to prevent Replicate from
```

```
scanning the log (to find the timestamp). When using IBM DB2 for LUW, 'Start from
 timestamp' is only supported if an actual
change was captured by this Replicate task earlier to the specified timestamp.
```

Résolution des problèmes de latence dans AWS Database Migration Service

Cette section fournit une vue d'ensemble des causes courantes de latence des tâches AWS DMS pendant la phase de réplication continue (CDC). AWS DMS réplique les données de manière asynchrone. La latence est le délai entre le moment où une modification est validée sur la source et le moment où la modification est répliquée sur la cible. La latence peut être due à une mauvaise configuration des composants de réplication, tels que :

- Point de terminaison source ou source de données
- Point de terminaison cible ou source de données
- Instances de réplication
- Le réseau entre ces composants

Nous vous recommandons d'utiliser un test de migration comme preuve de concept, afin de recueillir des informations sur la réplication. Vous pouvez ensuite utiliser ces informations pour ajuster votre configuration de réplication afin de réduire au maximum la latence. Pour en savoir plus sur l'exécution d'une migration faisant office de preuve de concept, consultez Exécution d'une preuve de concept.

Rubriques

- Types de latence de CDC
- Causes courantes de la latence de CDC
- Résolution des problèmes de latence

Types de latence de CDC

Cette section décrit les types de latence de réplication qui peuvent survenir pendant la CDC.

Latence source

Retard, en secondes, entre l'instant de validation du dernier événement capturé à partir du point de terminaison source et l'horodatage système actuel de l'instance de réplication. Vous pouvez

surveiller la latence entre la source de données et votre instance de réplication à l'aide de la `CDCLatencySource` CloudWatch métrique. Une métrique `CDCLatencySource` élevée indique que le processus de capture des modifications depuis la source est retardé. Par exemple, si votre application valide une insertion dans la source à 10 h 00 et qu'AWS DMS consomme cette modification à 10 h 02, la métrique `CDCLatencySource` est de 120 secondes.

Pour plus d'informations sur CloudWatch les métriques pourAWS DMS, voirMétriques de tâches de réplication.

Latence cible

Retard, en secondes, entre l'instant de validation sur la source du dernier événement en attente d'être validé sur la cible et l'horodatage actuel de l'instance de réplication DMS. Vous pouvez surveiller la latence entre les validations sur la source de données et votre cible de données à l'aide de la `CDCLatencyTarget` CloudWatch métrique. Cela signifie que `CDCLatencyTarget` inclut chaque retard de lecture depuis la source. Par conséquent, `CDCLatencyTarget` est toujours supérieur ou égal à `CDCLatencySource`.

Par exemple, si votre application valide une insertion dans la source à 10 h 00 et qu'AWS DMS la consomme à 10 h 02 et l'écrit dans la cible à 10 h 05, la métrique `CDCLatencyTarget` est de 300 secondes.

Causes courantes de la latence de CDC

Cette section décrit les causes de latence que la réplication peut rencontrer pendant la CDC.

Rubriques

- Ressources des points de terminaison
- Ressources d'instance de réplication
- Vitesse et bande passante du réseau
- Configuration DMS
- Scénarios de réplication

Ressources des points de terminaison

Les facteurs suivants ont une incidence significative sur les performances et la latence de réplication :

- Configurations des bases de données source et cible

- Taille d'instance
- Magasins de données source ou cible sous-provisionnés ou mal configurés

Pour identifier les causes de latence causées par des problèmes de point de AWS terminaison pour les sources et les cibles hébergées, surveillez les CloudWatch indicateurs suivants :

- `FreeMemory`
- `CPUUtilization`
- Métriques de débit et d'E/S, telles que `WriteIOPS`, `WriteThroughput` ou `ReadLatency`
- Métriques de volume de transactions telles que `CDCIncomingChanges`.

Pour plus d'informations sur CloudWatch les métriques de surveillance, consultezMétriques AWS Database Migration Service.

Ressources d'instance de réplication

Les ressources d'instance de réplication sont essentielles à la réplication, et vous devez vous assurer qu'il n'y a pas de goulots d'étranglement liés aux ressources, car ils peuvent entraîner une latence source et cible également.

Pour identifier les goulots d'étranglement de ressources pour votre instance de réplication, vérifiez les points suivants :

- CloudWatch Les indicateurs critiques tels que le processeur, la mémoire, les E/S par seconde et le stockage ne connaissent pas de pics ou de valeurs constamment élevées.
- La taille de l'instance de réplication est adaptée à la charge de travail. Pour en savoir plus sur la détermination de la taille correcte d'une instance de réplication, consultez Sélection de la meilleure taille pour une instance de réplication.

Vitesse et bande passante du réseau

La bande passante du réseau est un facteur qui affecte la transmission des données. Pour analyser les performances réseau de la réplication, effectuez l'une des opérations suivantes :

- Vérifiez les métriques `ReadThroughput` et `WriteThroughput` au niveau de l'instance. Pour plus d'informations sur CloudWatch les métriques de surveillance, consultezMétriques AWS Database Migration Service.

- Utilisez l'AMI d'assistance au diagnostic AWS DMS. Si l'AMI d'assistance au diagnostic n'est pas disponible dans votre région, vous pouvez la télécharger depuis n'importe quelle région prise en charge et la copier dans votre région pour effectuer l'analyse de votre réseau. Pour en savoir plus sur l'AMI d'assistance au diagnostic, consultez Utilisation de l'AMI d'assistance au diagnostic AWS DMS.

La fonctionnalité CDC dans AWS DMS est à thread unique pour garantir la cohérence des données. Par conséquent, vous pouvez déterminer le volume de données que votre réseau peut prendre en charge en calculant votre taux de transfert de données à thread unique. Par exemple, si la tâche se connecte à sa source via un réseau de 100 Mbit/s (mégabits par seconde), la réplication dispose d'une allocation de bande passante maximale théorique de 12,5 Mo/s (mégaoctets par seconde). Cela équivaut à 45 gigabits par heure. Si le taux de génération de journaux de transactions sur la source est supérieur à 45 gigabits par heure, cela signifie que la tâche a une latence de CDC. Pour un réseau de 100 Mo/s, ces taux sont des maxima théoriques ; d'autres facteurs tels que le trafic réseau et la surcharge de ressources sur la source et la cible réduisent la bande passante réellement disponible.

Configuration DMS

Cette section contient des configurations de réplication recommandées qui peuvent contribuer à réduire la latence.

- Paramètres des points de terminaison : les paramètres des points de terminaison sources et cibles peuvent nuire aux performances de votre instance de réplication. Les paramètres de point de terminaison qui activent des fonctionnalités gourmandes en ressources auront un impact sur les performances. Par exemple, pour un point de terminaison Oracle, la désactivation LogMiner et l'utilisation de Binary Reader améliorent les performances, car cela demande beaucoup LogMiner de ressources. Le paramètre de point de terminaison suivant améliore les performances d'un point de terminaison Oracle :

  ```
  useLogminerReader=N;useBfile=Y
  ```

  Pour plus d'informations sur les paramètres de point de terminaison, consultez la documentation relative au moteur des points de terminaison sources et cibles dans la rubrique Utilisation des points de terminaison AWS DMS.
- Paramètres des tâches : certains paramètres de tâche pour votre scénario de réplication spécifique peuvent nuire aux performances de votre instance de réplication. Par exemple, AWS DMS utilise

le mode d'application transactionnel par défaut (`BatchApplyEnabled=false`) pour la CDC pour tous les points de terminaison à l'exception d'Amazon Redshift. Toutefois, pour les sources comportant un grand nombre de modifications, le réglage de `BatchApplyEnabled` sur `true` peut améliorer les performances.

Pour plus d'informations sur les paramètres de tâche, consultez Spécification des paramètres des tâches pour les tâches du AWS Database Migration Service.

- Position de départ d'une tâche de CDC uniquement : le démarrage d'une tâche de CDC uniquement à partir d'une position ou d'un horodatage antérieurs entraîne le démarrage de la tâche avec une latence source de CDC accrue. En fonction du volume des modifications apportées à la source, la latence des tâches mettra du temps à s'atténuer.
- Paramètres LOB : les types de données LOB peuvent entraver les performances de réplication en raison de la manière dont AWS DMS réplique les données binaires volumineuses. Pour plus d'informations, consultez les rubriques suivantes :
  - Configuration du support LOB pour les bases de données sources dans une tâche AWS DMS
  - Migration des objets binaires volumineux (Large Binary Object, LOB).

Scénarios de réplication

Cette section décrit des scénarios de réplication spécifiques et la manière dont ils peuvent affecter la latence.

Rubriques

- Arrêt d'une tâche pour une période de temps prolongée
- Modifications mises en cache
- Réplication entre régions

Arrêt d'une tâche pour une période de temps prolongée

Lorsque vous arrêtez une tâche, AWS DMS enregistre la position du dernier journal de transactions lu depuis la source. Lorsque vous reprenez la tâche, DMS essaie de poursuivre la lecture à partir de la même position dans le journal des transactions. La reprise d'une tâche après plusieurs heures ou plusieurs jours entraîne l'augmentation de la latence source de CDC jusqu'à ce que DMS ait fini de consommer le backlog des transactions.

Modifications mises en cache

Les modifications mises en cache sont des modifications que votre application écrit dans la source de données pendant qu'AWS DMS exécute la phase de réplication à chargement complet. DMS n'applique pas ces modifications tant que la phase de chargement complet n'est pas terminée et que la phase de CDC ne démarre pas. Pour une source comportant un grand nombre de transactions, les modifications mises en cache mettent plus de temps à s'appliquer, de sorte que la latence source augmente lorsque la phase de CDC démarre. Nous vous recommandons d'exécuter la phase de chargement complet lorsque les volumes de transactions sont faibles afin de minimiser le nombre de modifications mises en cache.

Réplication entre régions

La localisation de vos points de terminaison DMS ou de votre instance de réplication dans différentes régions AWS augmente la latence réseau. Cela augmente la latence de réplication. Pour des performances optimales, localisez votre point de terminaison source, votre point de terminaison cible et votre instance de réplication dans la même région AWS.

Résolution des problèmes de latence

Cette section contient les étapes de résolution des problèmes de latence de réplication.

Pour résoudre un problème de latence, procédez comme suit :

- Commencez par déterminer le type et le degré de latence de la tâche. Consultez la section Statistiques de table de la tâche dans la console ou l'interface de ligne de commande DMS. Si les compteurs changent, la transmission des données est en cours. Vérifiez les métriques `CDCLatencySource` et `CDCLatencyTarget` ensemble pour déterminer s'il y a un goulot d'étranglement pendant la CDC.
- Si des métriques `CDCLatencySource` ou `CDCLatencyTarget` élevées indiquent la présence d'un goulot d'étranglement dans la réplication, vérifiez les points suivants :
  - Si la valeur `CDCLatencySource` est élevée et que `CDCLatencyTarget` a pour valeur `CDCLatencySource`, cela indique la présence d'un goulot d'étranglement dans le point de terminaison source et qu'AWS DMS écrit des données dans la cible de manière fluide. Consultez Résolution des problèmes de latence source ci-dessous.
  - Si la valeur `CDCLatencySource` est faible et la valeur `CDCLatencyTarget` élevée, cela indique la présence d'un goulot d'étranglement dans votre point de terminaison cible et qu'AWS DMS lit des données de la source de manière fluide. Consultez Résolution des problèmes de latence cible ci-dessous.

- Si la valeur `CDCLatencySource` est élevée et que la valeur `CDCLatencyTarget` est nettement supérieure à `CDCLatencySource`, cela indique des goulots d'étranglement à la fois pour les lectures sources et les écritures cibles. Commencez par étudier la latence source, puis étudiez la latence cible.

Pour en savoir plus sur la surveillance des métriques des tâches DMS, consultez Surveillance des tâches AWS DMS.

Résolution des problèmes de latence source

Les rubriques suivantes décrivent des scénarios de réplication spécifiques aux types de point de terminaison source.

Rubriques

- Résolution de problèmes de point de terminaison Oracle
- Résolution des problèmes liés aux points de terminaison MySQL
- Résolution des problèmes liés aux points de terminaison PostgreSQL
- Résolution des problèmes liés aux points de terminaison SQL Server

Résolution de problèmes de point de terminaison Oracle

Cette section contient des scénarios de réplication spécifiques à Oracle.

Lecture source interrompue

AWS DMS interrompt la lecture depuis une source Oracle dans les scénarios suivants. Ce comportement est intégré à la conception. Vous pouvez en rechercher les causes à l'aide du journal des tâches. Recherchez les messages similaires aux suivants dans le journal des tâches. Pour en savoir plus sur l'utilisation du journal des tâches, consultez Affichage et gestion des journaux de tâches AWS DMS.

- Message SORTER : indique que DMS met en cache les transactions sur l'instance de réplication. Pour plus d'informations, consultez Message SORTER dans le journal des tâches, ci-après.
- Journaux des tâches de débogage : si DMS interrompt le processus de lecture, votre tâche écrit à plusieurs reprises le message suivant dans les journaux des tâches de débogage, sans modifier le champ de contexte ni l'horodatage :
  - Binary Reader :

```
[SOURCE_CAPTURE  ]T:  Produce CTI event:
context '00000020.f23ec6e5.00000002.000a.00.0000:190805.3477731.16'
xid [00000000001e0018] timestamp '2021-07-19 06:57:55'
thread 2  (oradcdc_oralog.c:817)
```

- Logminer :

```
[SOURCE_CAPTURE  ]T:  Produce INSERT event:
object id 1309826 context
 '000000000F2CECAA010000010005A8F500000275016C0000000000000F2CEC58'
xid [000014e06411d996] timestamp '2021-08-12 09:20:32' thread 1
 (oracdc_reader.c:2269)
```

- AWS DMS enregistre le message suivant pour chaque nouvelle opération des journaux redo ou d'archivage.

```
00007298: 2021-08-13T22:00:34 [SOURCE_CAPTURE ]I: Start processing archived
 Redo log sequence 14850 thread 2 name XXXXX/XXXXX/ARCHIVELOG/2021_08_14/
thread_2_seq_14850.22977.1080547209 (oradcdc_redo.c:754)
```

  Si la source dispose de nouvelles opérations de journaux redo ou d'archivage, et qu'AWS DMS n'écrit pas ces messages dans le journal, cela signifie que la tâche ne traite pas les événements.

Haute génération de journaux redo

Si votre tâche traite des journaux redo ou d'archivage, mais que la latence source reste élevée, essayez d'identifier le taux de génération de journaux redo et les modèles de génération. Si le niveau de génération de journaux redo est élevé, cela augmente la latence source, car votre tâche lit tous les journaux redo et d'archivage afin d'extraire les modifications relatives aux tables répliquées.

Pour déterminer le taux de génération de journaux redo, utilisez les requêtes suivantes.

- Taux de génération de journaux redo par jour :

```
select trunc(COMPLETION_TIME,'DD') Day, thread#,
round(sum(BLOCKS*BLOCK_SIZE)/1024/1024/1024) GB,
count(*) Archives_Generated from v$archived_log
where completion_time > sysdate- 1
group by trunc(COMPLETION_TIME,'DD'),thread# order by 1;
```

- Taux de génération de journaux redo par heure :

```
Alter session set nls_date_format = 'DD-MON-YYYY HH24:MI:SS';
select trunc(COMPLETION_TIME,'HH') Hour,thread# ,
round(sum(BLOCKS*BLOCK_SIZE)/1024/1024) "REDO PER HOUR (MB)",
count(*) Archives from v$archived_log
where completion_time > sysdate- 1
group by trunc(COMPLETION_TIME,'HH'),thread#  order by 1 ;
```

Pour résoudre les problèmes de latence dans ce scénario, procédez aux vérifications suivantes :

- Vérifiez la bande passante du réseau et les performances mono-thread de la réplication pour vous assurer que votre réseau sous-jacent peut prendre en charge le taux de génération de journaux redo de la source. Pour en savoir plus sur la manière dont la bande passante du réseau peut affecter les performances de réplication, consultez Vitesse et bande passante du réseau précédemment.
- Vérifiez si vous avez correctement configuré la journalisation supplémentaire. Évitez une journalisation supplémentaire sur la source, comme l'activation de la journalisation sur toutes les colonnes d'une table. Pour en savoir plus sur la configuration d'une journalisation supplémentaire, consultez Configurez une journalisation supplémentaire.
- Vérifiez que vous utilisez l'API appropriée pour lire les journaux redo et d'archivage. Vous pouvez utiliser Oracle LogMiner ou AWS DMS Binary Reader. Tout en LogMiner lisant les journaux de journalisation en ligne et les fichiers de journalisation archivés, Binary Reader lit et analyse directement les fichiers de journalisation bruts. Par conséquent, Binary Reader est plus performant. Nous vous recommandons d'utiliser Binary Reader si votre génération de journaux redo dépasse 10 Go/heure. Pour plus d'informations, consultez Utilisation d'Oracle LogMiner ou de AWS DMS Binary Reader pour le CDC.
- Vérifiez si vous avez défini `ArchivedLogsOnly` sur `Y`. Si ce paramètre de point de terminaison est défini, AWS DMS lit les journaux redo archivés. Cela augmente la latence source, car AWS DMS attend que le journal redo en ligne soit archivé avant de le lire. Pour plus d'informations, consultez ArchivedLogsOnly.
- Si votre source Oracle utilise Automatic Storage Management (ASM), consultez Stockage de REDO sur Oracle ASM lors de l'utilisation d'Oracle comme source pour AWS DMS pour en savoir plus sur la manière de configurer correctement votre magasin de données. Vous pouvez également optimiser davantage les performances de lecture en utilisant l'attribut de connexion supplémentaire (ECA) `asmUsePLSQLArray`. Pour obtenir des informations sur l'utilisation d'`asmUsePLSQLArray`,

veuillez consulter Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS.

Résolution des problèmes liés aux points de terminaison MySQL

Cette section contient des scénarios de réplication spécifiques à MySQL. AWS DMS analyse régulièrement le journal binaire MySQL pour répliquer les modifications. Ce processus peut augmenter la latence dans les scénarios suivants :

Rubriques

- Transaction de longue durée sur la source
- Charge de travail élevée sur la source
- Contention des journaux binaires

Transaction de longue durée sur la source

Comme MySQL écrit uniquement les transactions validées dans le journal binaire, les transactions de longue durée provoquent des pics de latence proportionnels à la durée d'exécution des requêtes.

Pour identifier les transactions de longue durée, utilisez la requête suivante ou utilisez le journal des requêtes lentes :

```
SHOW FULL PROCESSLIST;
```

Pour plus d'informations sur l'utilisation du journal des requêtes lentes, consultez Journal des requêtes lentes (langue française non garantie) dans la documentation MySQL.

Pour éviter les pics de latence liés aux transactions de longue durée, restructurez vos transactions sources afin de réduire le temps d'exécution des requêtes ou d'augmenter la fréquence de validation.

Charge de travail élevée sur la source

La CDC DMS étant monothread, un grand nombre de transactions peut augmenter la latence source. Pour déterminer si la latence source est due à une charge de travail importante, comparez le nombre et la taille des journaux binaires générés pendant la période de latence aux journaux générés avant cette latence. Pour vérifier les journaux binaires et le statut du thread de CDC DMS, utilisez les requêtes suivantes :

```
SHOW BINARY LOGS;
SHOW PROCESSLIST;
```

Pour plus d'informations sur les états des threads de vidage de journaux binaires de CDC, consultez États des threads de la source de réplication (langue française non garantie).

Vous pouvez déterminer la latence en comparant la dernière position du journal binaire générée sur la source avec l'événement en cours de traitement par DMS. Pour identifier le dernier journal binaire de la source, procédez comme suit :

- Activez les journaux de débogage sur le composant SOURCE_CAPTURE.
- Récupérez le journal binaire de traitement DMS et les détails de position dans les journaux de débogage des tâches.
- Utilisez la requête suivante pour identifier le dernier journal binaire de la source :

```
SHOW MASTER STATUS;
```

Pour optimiser davantage les performances, réglez `EventsPollInterval`. Par défaut, DMS interroge le journal binaire toutes les 5 secondes, mais vous pouvez améliorer les performances en réduisant cette valeur. Pour plus d'informations sur le paramètre `EventsPollInterval`, consultez Paramètres de point de terminaison lors de l'utilisation de MySQL en tant que source pour AWS DMS.

Contention des journaux binaires

Lorsque vous migrez plusieurs tables contenant une grande quantité de données, nous vous recommandons de les diviser en tâches distinctes pour MySQL 5.7.2 ou version ultérieure. Dans MySQL versions 5.7.2 et ultérieures, le thread de vidage principal réduit le nombre de conflits de verrouillage et améliore le débit. Par conséquent, le thread de vidage ne verrouille plus le journal binaire chaque fois qu'il lit un événement. Cela signifie que plusieurs threads de vidage peuvent lire le fichier journal binaire simultanément. Cela signifie également que les threads de vidage peuvent lire le journal binaire pendant que les clients écrivent dans celui-ci. Pour plus d'informations sur les threads de vidage, consultez Threads de réplication (langue française non garantie) et les notes de mise à jour de MySQL 5.7.2 (langue française non garantie).

Pour améliorer les performances de réplication pour les versions sources MySQL antérieures à 5.7.2, essayez de consolider les tâches avec les composants CDC.

Résolution des problèmes liés aux points de terminaison PostgreSQL

Cette section contient des scénarios de réplication spécifiques à PostgreSQL.

Rubriques

- Transaction de longue durée sur la source
- Charge de travail élevée sur la source
- Débit réseau élevé
- Déversez des fichiers dans Aurora PostgreSQL

Transaction de longue durée sur la source

Lorsque la base de données source contient des transactions de longue durée, telles que des milliers d'insertions dans une seule transaction, les compteurs de transactions et d'événements CDC DMS n'augmentent pas tant que la transaction n'est pas terminée. Ce retard peut entraîner des problèmes de latence que vous pouvez mesurer à l'aide de la métrique `CDCLatencyTarget`.

Pour examiner les transactions de longue durée, effectuez l'une des opérations suivantes :

- Utilisez la vue `pg_replication_slots`. Si la valeur `restart_lsn` n'est pas mise à jour, il est probable que PostgreSQL ne soit pas en mesure de publier les journaux d'écriture anticipée (WAL) en raison de transactions actives de longue durée. Pour en savoir plus sur la vue `pg_replication_slots`, consultez pg_replication_slots dans la documentation de PostgreSQL 15.4.
- Utilisez la requête suivante pour renvoyer la liste de toutes les requêtes actives dans la base de données, ainsi que les informations associées :

```
SELECT pid, age(clock_timestamp(), query_start), usename, query
FROM pg_stat_activity WHERE query != '<IDLE>'
AND query NOT ILIKE '%pg_stat_activity%'
ORDER BY query_start desc;
```

  Dans les résultats de la requête, le champ `age` indique la durée active de chaque requête, que vous pouvez utiliser pour identifier les requêtes de longue durée.

Charge de travail élevée sur la source

Si la charge de travail de votre source PostgreSQL est élevée, vérifiez les points suivants pour réduire la latence :

- Vous pouvez rencontrer une latence élevée lorsque vous utilisez le plug-in `test_decoding` lors de la migration d'un sous-ensemble de tables de la base de données source avec une valeur élevée de transactions par seconde (TPS). Cela est dû au fait que le plug-in `test_decoding` envoie toutes les modifications de base de données à l'instance de réplication, que DMS filtre ensuite en fonction du mappage de table de la tâche. Les événements relatifs aux tables qui ne font pas partie du mappage des tables de la tâche peuvent augmenter la latence source.
- Vérifiez le débit TPS en utilisant l'une des méthodes suivantes.
  - Pour les sources Aurora PostgreSQL, utilisez la métrique. `CommitThroughput` CloudWatch
  - Pour PostgreSQL qui s'exécute sur Amazon RDS ou sur site, utilisez la requête suivante à l'aide d'un client PSQL version 11 ou ultérieure (appuyez sur **`enter`** pendant la requête pour faire avancer les résultats) :

    ```
    SELECT SUM(xact_commit)::numeric as temp_num_tx_ini FROM pg_stat_database; \gset
    select pg_sleep(60);
    SELECT SUM(xact_commit)::numeric as temp_num_tx_final FROM pg_stat_database; \gset
    select (:temp_num_tx_final - :temp_num_tx_ini)/ 60.0 as "Transactions Per Second";
    ```
- Pour réduire la latence lors de l'utilisation du plug-in `test_decoding`, envisagez plutôt d'utiliser le plug-in `pglogical`. Contrairement au plug-in `test_decoding`, le plug-in `pglogical` filtre les modifications du journal d'écriture anticipée (WAL) à la source et envoie uniquement les modifications pertinentes à l'instance de réplication. Pour en savoir plus sur l'utilisation du plug-in `pglogical` avec AWS DMS, consultez Configuration du plug-in pglogical.

Débit réseau élevé

La réplication peut utiliser beaucoup de bande passante du réseau lorsque vous utilisez le plug-in `test_decoding`, en particulier lors de transactions à volume élevé. Cela est dû au fait que le plug-in `test_decoding` traite les modifications et les convertit dans un format lisible par l'homme qui est plus grand que le format binaire d'origine.

Pour améliorer les performances, pensez plutôt à utiliser le plug-in `pglogical`, qui est un plug-in binaire. Contrairement au plug-in `test_decoding`, le plug-in `pglogical` génère une sortie au format binaire, ce qui entraîne des modifications du flux WAL (Write Ahead Log) compressé.

Déversez des fichiers dans Aurora PostgreSQL

Dans les versions 13 et supérieures de PostgreSQL, `logical_decoding_work_mem` le paramètre détermine l'allocation de mémoire pour le décodage et le streaming. Pour plus d'informations sur le `logical_decoding_work_mem` paramètre, consultez la section Consommation de ressources dans PostgreSQL dans la documentation de PostgreSQL 13.13.

La réplication logique accumule les modifications pour toutes les transactions en mémoire jusqu'à ce que ces transactions soient validées. Si la quantité de données stockée pour toutes les transactions dépasse la quantité spécifiée par le paramètre de base de données`logical_decoding_work_mem`, DMS déverse les données de transaction sur le disque afin de libérer de la mémoire pour les nouvelles données de décodage.

Les transactions de longue durée, ou de nombreuses sous-transactions, peuvent entraîner une consommation accrue de mémoire de décodage logique par le DMS. Cette augmentation de l'utilisation de la mémoire entraîne la création par DMS de fichiers indésirables sur le disque, ce qui entraîne une latence élevée de la source lors de la réplication.

Pour réduire l'impact d'une augmentation de la charge de travail source, procédez comme suit :

- Réduisez les transactions de longue durée.
- Réduisez le nombre de sous-transactions.
- Évitez d'effectuer des opérations qui génèrent une grande quantité d'enregistrements de journal, telles que la suppression ou la mise à jour d'une table entière en une seule transaction. Effectuez plutôt des opérations par petits lots.

Vous pouvez utiliser les CloudWatch mesures suivantes pour surveiller la charge de travail sur la source :

- `TransactionLogsDiskUsage`: le nombre d'octets actuellement occupés par le WAL logique. Cette valeur augmente de façon monotone si les emplacements de réplication logiques ne sont pas en mesure de suivre le rythme des nouvelles écritures ou si des transactions de longue durée empêchent le ramassage des anciens fichiers.
- `ReplicationSlotDiskUsage`: quantité d'espace disque actuellement utilisée par les emplacements de réplication logique.

Vous pouvez réduire la latence de la source en ajustant le `logical_decoding_work_mem` paramètre. La valeur par défaut de ce paramètre est de 64 Mo. Ce paramètre limite la quantité de

mémoire utilisée par chaque connexion de réplication en continu logique. Nous recommandons de définir `logical_decoding_work_mem` une valeur nettement supérieure à la `work_mem` valeur afin de réduire le nombre de modifications décodées que le DMS écrit sur le disque.

Nous vous recommandons de vérifier régulièrement la présence de fichiers déversés, en particulier pendant les périodes de forte activité de migration ou de latence. Si le DMS crée un nombre important de fichiers de déversement, cela signifie que le décodage logique ne fonctionne pas efficacement, ce qui peut augmenter le temps de latence. Pour atténuer ce problème, augmentez la valeur du `logical_decoding_work_mem` paramètre.

Vous pouvez vérifier le dépassement des transactions en cours à l'aide de la `aurora_stat_file` fonction. Pour plus d'informations, consultez la section Ajustement de la mémoire de travail pour le décodage logique dans le manuel Amazon Relational Database Service Developer Guide.

Résolution des problèmes liés aux points de terminaison SQL Server

Cette section contient des scénarios de réplication spécifiques à SQL Server. Pour déterminer les modifications à répliquer à partir de SQL Server, AWS DMS lit les journaux de transactions et effectue des analyses périodiques sur la base de données source. La latence de réplication résulte généralement du fait que SQL Server limite ces analyses en raison de contraintes sur les ressources. Elle peut également résulter d'une augmentation significative du nombre d'événements écrits dans le journal des transactions en peu de temps.

Rubriques

- Reconstructions d'index
- Transactions volumineuses
- Intervalle d'interrogation MS-CDC mal configuré pour Amazon RDS SQL Server
- Réplication de plusieurs tâches de CDC à partir de la même base de données source

Reconstructions d'index

Lorsque SQL Server reconstruit un index volumineux, il utilise une seule transaction. Cela génère de nombreux événements et peut utiliser une grande quantité d'espace de journalisation si SQL Server reconstruit plusieurs index à la fois. Quand cela se produit, vous pouvez vous attendre à de brefs pics de réplication. Si votre source SQL Server présente des pics de journalisation persistants, vérifiez les points suivants :

- Vérifiez d'abord la durée des pics de latence à l'aide des `CDCLatencySource` CloudWatch métriques `CDCLatencySource` et, ou en consultant les messages de surveillance du débit dans les journaux des tâches. Pour plus d'informations sur CloudWatch les métriques pourAWS DMS, voirMétriques de tâches de réplication.
- Vérifiez si la taille des journaux de transactions actifs ou des sauvegardes de journaux a augmenté au cours du pic de latence. Vérifiez également si une tâche de maintenance ou une reconstruction a été exécutée pendant cette période. Pour en savoir plus sur la vérification de la taille des journaux de transactions, consultez Surveiller l'utilisation de l'espace pour le journal dans la documentation technique de SQL Server.
- Vérifiez que votre plan de maintenance respecte les bonnes pratiques relatives à SQL Server. Pour en savoir plus sur les bonnes pratiques de maintenance de SQL Server, consultez Stratégie de maintenance d'index dans la documentation technique de SQL Server.

Pour résoudre les problèmes de latence lors des reconstructions d'index, essayez ce qui suit :

- Utilisez le modèle de récupération `BULK_LOGGED` pour les reconstructions hors connexion afin de réduire le nombre d'événements qu'une tâche doit traiter.
- Si possible, arrêtez la tâche pendant les reconstructions d'index. Vous pouvez également essayer de planifier des reconstructions d'index en dehors des heures de pointe pour atténuer l'impact d'un pic de latence.
- Essayez d'identifier les goulots d'étranglement de ressources qui ralentissent les lectures DMS, tels que la latence du disque ou le débit d'E/S, et résolvez-les.

Transactions volumineuses

Les transactions comportant un grand nombre d'événements, ou les transactions de longue durée, font grossir le journal des transactions.. Les lectures DMS prennent donc plus de temps, ce qui entraîne une latence. Cela est similaire à l'effet des reconstructions d'index sur les performances de réplication.

Vous pouvez avoir des difficultés à identifier ce problème si vous ne connaissez pas la charge de travail typique de la base de données source. Pour résoudre ce problème, procédez comme suit :

- Tout d'abord, identifiez le temps pendant lequel le temps de latence a augmenté à l'aide `WriteThroughput` CloudWatch des métriques `ReadThroughput` et, ou en consultant les messages de surveillance du débit dans les journaux des tâches.

- Vérifiez la présence ou non de requêtes de longue durée sur la base de données source pendant le pic de latence. Pour en savoir plus sur les requêtes de longue durée, consultez Résoudre les problèmes de lenteur des requêtes dans SQL Server dans la documentation technique de SQL Server.
- Vérifiez si la taille des journaux de transactions actifs ou des sauvegardes de journaux a augmenté. Pour plus d'informations, consultez Surveiller l'utilisation de l'espace pour le journal dans la documentation technique de SQL Server.

Pour résoudre ce problème, effectuez l'une des actions suivantes :

- La meilleure solution consiste à restructurer vos transactions côté application afin qu'elles se terminent rapidement.
- Si vous ne pouvez pas restructurer vos transactions, une solution de contournement à court terme consiste à rechercher des goulots d'étranglement liés aux ressources, tels que des temps d'attente sur le disque ou des conflits de CPU. Si vous trouvez des goulots d'étranglement dans la base de données source, vous pouvez réduire la latence en augmentant les ressources de disque, de CPU et de mémoire pour la base de données source. Cela réduit les conflits pour les ressources système, ce qui permet aux requêtes DMS de se terminer plus rapidement.

Intervalle d'interrogation MS-CDC mal configuré pour Amazon RDS SQL Server

Un paramètre d'intervalle d'interrogation mal configuré sur les instances Amazon RDS peut faire grossir le journal des transactions. Cela est dû au fait que la réplication empêche la troncation du journal. Bien que les tâches en cours d'exécution puissent continuer à se répliquer avec une latence minimale, l'arrêt et la reprise des tâches, ou le démarrage de tâches de CDC uniquement, peuvent entraîner l'échec des tâches. Cela est dû à des délais d'attente lors de l'analyse du journal des transactions volumineux.

Pour résoudre un intervalle d'interrogation mal configuré, procédez comme suit :

- Vérifiez si la taille du journal des transactions actif augmente et si l'utilisation du journal est proche de 100 %. Pour plus d'informations, consultez Surveiller l'utilisation de l'espace pour le journal dans la documentation technique de SQL Server.
- Vérifiez si la troncation du journal est retardée avec un paramètre `log_reuse_wait_desc value` égal à `REPLICATION`. Pour plus d'informations, consultez Le journal des transactions (SQL Server) dans la documentation technique de SQL Server.

Si vous rencontrez des problèmes avec l'un des éléments de la liste précédente, ajustez l'intervalle d'interrogation MS-CDC. Pour en savoir plus sur l'ajustement de l'intervalle d'interrogation, consultez Paramètres recommandés lors de l'utilisation d'Amazon RDS for SQL Server comme source pour AWS DMS.

Réplication de plusieurs tâches de CDC à partir de la même base de données source

Pendant la phase de chargement complet, nous recommandons de répartir les tables entre les tâches afin d'améliorer les performances, de séparer les tables dépendantes de manière logique et d'atténuer l'impact de l'échec d'une tâche. Toutefois, pendant la phase CDC, nous recommandons de consolider les tâches afin de minimiser les analyses DMS. Pendant la phase CDC, chaque tâche DMS analyse les journaux de transactions pour détecter de nouveaux événements plusieurs fois par minute. Comme chaque tâche s'exécute indépendamment, chaque tâche analyse chaque journal de transactions individuellement. Cela augmente l'utilisation de disque et de CPU sur la base de données SQL Server source. Par conséquent, l'exécution d'un grand nombre de tâches en parallèle peut entraîner une limitation des lectures DMS par SQL Server, ce qui augmente la latence.

Vous aurez peut-être du mal à identifier ce problème si plusieurs tâches démarrent progressivement. Le symptôme le plus courant de ce problème est que la plupart des analyses de tâches commencent à prendre plus de temps. Cela entraîne une latence plus élevée pour ces analyses. SQL Server donne la priorité à certaines analyses de tâches, de sorte que certaines tâches présentent une latence normale. Pour résoudre ce problème, vérifiez la métrique `CDCLatencySource` pour toutes vos tâches. Si `CDCLatencySource` augmente pour certaines tâches, alors que `CDCLatencySource` est faible pour d'autres, il est probable que SQL Server limite vos lectures DMS pour certaines de vos tâches.

Si SQL Server limite les lectures de vos tâches pendant la CDC, consolidez vos tâches afin de minimiser le nombre d'analyses DMS. Le nombre maximal de tâches pouvant se connecter à la base de données source sans créer de conflits dépend de facteurs tels que la capacité de la base de données source, le taux de croissance du journal des transactions ou le nombre de tables. Pour déterminer le nombre idéal de tâches pour votre scénario de réplication, testez la réplication dans un environnement de test similaire à votre environnement de production.

Résolution des problèmes de latence cible

Cette section contient des scénarios qui peuvent affecter la latence cible.

Rubriques

- Indexation des problèmes

- Message SORTER dans le journal des tâches
- Verrouillage de base de données
- Recherches d'objets LOB lentes
- Multi-AZ, journalisation des audits et sauvegardes

Indexation des problèmes

Pendant la phase CDC, AWS DMS réplique les modifications sur la source en exécutant des instructions DML (d'insertion, de mise à jour et de suppression) sur la cible. Pour les migrations hétérogènes à l'aide de DMS, des différences d'optimisation d'index sur la source et la cible peuvent entraîner un ralentissement des écritures sur la cible. Cela entraîne des problèmes de performances et de latence cible.

Pour résoudre des problèmes d'indexation, procédez comme suit. Les procédures relatives à ces étapes varient selon les moteurs de base de données.

- Surveillez la durée d'interrogation pour la base de données cible. La comparaison de la durée d'exécution des requêtes sur la cible et sur la source peut indiquer les index qu'il convient d'optimiser.
- Activez la journalisation pour les requêtes lentes.

Pour résoudre les problèmes d'indexation liés à des réplications de longue durée, procédez comme suit :

- Ajustez les index de vos bases de données source et cible afin que la durée d'exécution des requêtes soit similaire sur la source et sur la cible.
- Comparez les index secondaires utilisés dans les requêtes DML pour la source et la cible. Assurez-vous que les performances DML sur la cible soient comparables ou supérieures aux performances DML sur la source.

Notez que la procédure d'optimisation des index est spécifique à votre moteur de base de données. Il n'existe aucune fonctionnalité DMS permettant d'ajuster les index source et cible.

Message SORTER dans le journal des tâches

Si un point de terminaison cible ne parvient pas à suivre le volume de modifications que AWS DMS y écrit, la tâche met en cache les modifications sur l'instance de réplication. Si le cache croît et

dépasse un seuil interne, la tâche arrête de lire les modifications provenant de la source. DMS procède ainsi pour éviter que l'instance de réplication ne soit à court de stockage ou que la tâche ne soit bloquée lors de la lecture d'un grand volume d'événements en attente.

Pour résoudre ce problème, consultez les CloudWatch journaux pour y trouver un message similaire à l'un des suivants :

```
[SORTER ]I: Reading from source is paused. Total disk usage exceeded the limit 90%
 (sorter_transaction.c:110)
[SORTER ]I: Reading from source is paused. Total storage used by swap files exceeded
 the limit 1048576000 bytes  (sorter_transaction.c:110)
```

Si vos journaux contiennent un message similaire au premier message, désactivez tout enregistrement de suivi pour la tâche et augmentez le stockage des instances de réplication. Pour en savoir plus sur l'augmentation du stockage des instances de réplication, consultez Modification d'une instance de réplication.

Si vos journaux contiennent un message similaire au second message, procédez comme suit :

- Déplacez les tables contenant de nombreuses transactions ou des opérations DML de longue durée vers une tâche distincte, si elles n'ont pas de dépendances sur les autres tables de la tâche.
- Augmentez les paramètres `MemoryLimitTotal` et `MemoryKeepTime` pour conserver la transaction plus longtemps en mémoire. Cela n'aidera pas si la latence est maintenue, mais cela peut aider à maintenir une latence faible pendant de courtes périodes intensives de volume transactionnel. Pour en savoir plus sur ces paramètres de tâche, consultez Paramètres de réglage du traitement des modifications.
- Évaluez si vous pouvez utiliser l'application par lots pour votre transaction en définissant `BatchApplyEnabled` sur `true`. Pour en savoir plus sur le paramètre `BatchApplyEnabled`, consultez Paramètres de métadonnées des tâches cibles.

Verrouillage de base de données

Si une application accède à une base de données que AWS DMS utilise en tant que cible de réplication, elle peut verrouiller une table à laquelle DMS essaie d'accéder. Cela crée un conflit de verrouillage. Comme DMS écrit les modifications dans la base de données cible dans l'ordre où elles se sont produites dans la source, les retards d'écriture dans une table dus à des conflits de verrouillage créent des retards d'écriture dans toutes les tables.

Pour résoudre ce problème, interrogez la base de données cible pour vérifier si un conflit de verrouillage bloque les transactions d'écriture DMS. Si la base de données cible bloque les transactions d'écriture DMS, effectuez une ou plusieurs des opérations suivantes :

- Restructurez vos requêtes pour valider les modifications plus fréquemment.
- Modifiez vos paramètres de délai de verrouillage.
- Partitionnez vos tables pour minimiser les conflits de verrouillage.

Notez que la procédure d'optimisation des conflits de verrouillage est spécifique à votre moteur de base de données. Il n'existe aucune fonctionnalité DMS permettant de régler les conflits de verrouillage.

Recherches d'objets LOB lentes

Quand AWS DMS réplique une colonne d'objets volumineux (LOB), il effectue une recherche sur la source juste avant d'écrire les modifications sur la cible. Cette recherche n'entraîne normalement aucune latence sur la cible, mais si la base de données source retarde la recherche en raison d'un verrouillage, la latence cible peut augmenter.

Ce problème est généralement difficile à diagnostiquer. Pour résoudre ce problème, activez le débogage détaillé sur les journaux de tâches et comparez les horodatages des appels de recherche d'objets LOB par DMS. Pour en savoir plus sur l'activation du débogage détaillé, consultez Affichage et gestion des journaux de tâches AWS DMS.

Pour résoudre ce problème, essayez les opérations suivantes :

- Améliorez les performances des requêtes SELECT sur la base de données source.
- Réglez les paramètres LOB dans DMS. Pour en savoir plus sur le réglage des paramètres LOB, consultez Migration des objets binaires volumineux (Large Binary Object, LOB).

Multi-AZ, journalisation des audits et sauvegardes

Pour les cibles Amazon RDS, la latence cible peut augmenter dans les cas suivants :

- Sauvegardes
- Après avoir activé plusieurs zones de disponibilité (multi-AZ)
- Après avoir activé la journalisation de la base de données, telle que les journaux d'audit ou de requêtes lentes.

Ces problèmes sont généralement difficiles à diagnostiquer. Pour résoudre ces problèmes, surveillez la latence pour détecter les pics périodiques pendant les fenêtres de maintenance Amazon RDS ou pendant les périodes de fortes charges de base de données.

Pour résoudre ces problèmes, essayez les opérations suivantes :

- Si possible, lors d'une migration à court terme, désactivez le mode multi-AZ, les sauvegardes ou la journalisation.
- Replanifiez vos fenêtres de maintenance pour les périodes de faible activité.

Utilisation de scripts d'assistance au diagnostic dans AWS DMS

Si vous rencontrez un problème lors de l'utilisation d'AWS DMS, votre ingénieur support peut avoir besoin de plus d'informations sur la base de données source ou cible. Nous voulons nous assurer qu'AWS Support obtienne autant d'informations que possible dans les plus brefs délais. À cet effet, nous avons développé des scripts pour interroger ces informations pour plusieurs des principaux moteurs de base de données relationnelle.

Si un script d'assistance est disponible pour la base de données, vous pouvez le télécharger à l'aide du lien figurant dans la rubrique du script correspondant décrite ci-dessous. Après avoir vérifié et examiné le script (décrit ci-dessous), vous pouvez l'exécuter conformément à la procédure décrite dans la rubrique du script. Lorsque l'exécution du script est terminée, vous pouvez charger sa sortie dans votre dossier AWS Support (encore une fois, comme décrit ci-dessous).

Avant d'exécuter le script, vous pouvez détecter les erreurs susceptibles d'avoir été introduites lors du téléchargement ou du stockage du script d'assistance. Pour ce faire, comparez la somme de contrôle du fichier de script à une valeur fournie par AWS. AWS utilise l'algorithme SHA256 pour la somme de contrôle.

Pour vérifier le fichier de script d'assistance à l'aide d'une somme de contrôle

1. Ouvrez le dernier fichier de somme de contrôle fourni pour vérifier ces scripts d'assistance à l'adresse https://d2pwp9zz55emqw.cloudfront.net/sha256Check.txt. Par exemple, le contenu du fichier peut être le suivant.

```
MYSQL  dfafd0d511477c699f96c64693ad0b1547d47e74d5c5f2f2025b790b1422e3c8
ORACLE  6c41ebcfc99518cfa8a10cb2ce8943b153b2cc7049117183d0b5de3d551bc312
POSTGRES  6ccd274863d14f6f3146fbdbbba43f2d8d4c6a4c25380d7b41c71883aa4f9790
```

```
SQL_SERVER  971a6f2c46aec8d083d2b3b6549b1e9990af3a15fe4b922e319f4fdd358debe7
```

2. Exécutez la commande de validation SHA256 pour votre système d'exploitation dans le répertoire qui contient le fichier d'assistance. Par exemple, sur le système d'exploitation macOS, vous pouvez exécuter la commande suivante sur un script d'assistance Oracle décrit plus loin dans cette rubrique.

```
shasum -a 256 awsdms_support_collector_oracle.sql
```

3. Comparez les résultats de la commande avec la valeur indiquée dans le dernier fichier `sha256Check.txt` que vous avez ouvert. Les deux valeurs doivent être identiques. Si ce n'est pas le cas, contactez votre ingénieur support pour en savoir plus sur la non-correspondance et sur la manière d'obtenir un fichier de script d'assistance propre.

Si vous disposez d'un fichier de script d'assistance propre, avant d'exécuter le script, veillez à lire et comprendre le code SQL du point de vue des performances et de la sécurité. Si vous n'êtes pas à l'aise avec l'exécution du code SQL dans ce script, vous pouvez mettre en commentaires ou supprimer le code SQL problématique. Vous pouvez également consulter votre ingénieur support pour connaître les solutions de contournement acceptables.

En cas d'exécution réussie et sauf indication contraire, le script renvoie la sortie dans un format HTML lisible. Le script est conçu pour exclure de ce contenu HTML toutes les données ou informations de sécurité susceptibles de compromettre votre activité. De plus, il n'apporte aucune modification à la base de données ou à son environnement. Toutefois, si vous trouvez dans le contenu HTML des informations que vous ne souhaitez pas partager, n'hésitez pas à supprimer les informations problématiques avant de charger le contenu HTML. Lorsque le contenu HTML est acceptable, chargez-le en utilisant les pièces jointes dans Détails du cas de votre dossier de support.

Chacune des rubriques suivantes décrit les scripts disponibles pour une base de données AWS DMS prise en charge et explique comment les exécuter. Votre ingénieur support vous dirigera vers un script spécifique documenté ci-dessous.

Rubriques

- Scripts d'assistance au diagnostic Oracle
- Scripts d'assistance au diagnostic SQL Server
- Scripts d'assistance au diagnostic pour les bases de données compatibles MySQL
- Scripts d'assistance au diagnostic PostgreSQL

Scripts d'assistance au diagnostic Oracle

Vous trouverez ci-dessous les scripts d'assistance au diagnostic disponibles pour analyser une base de données sur site ou Amazon RDS for Oracle dans votre configuration de migration AWS DMS. Ces scripts fonctionnent avec un point de terminaison source ou cible. Les scripts sont tous écrits pour être exécutés dans l'utilitaire de ligne de commande SQL*Plus. Pour plus d'informations sur l'utilisation de cet utilitaire, consultez A Utilisation de la ligne de commande SQL (langue française non garantie) dans la documentation Oracle.

Avant d'exécuter le script, veillez à ce que le compte d'utilisateur que vous utilisez dispose des autorisations nécessaires pour accéder à la base de données Oracle. Les paramètres d'autorisation affichés supposent qu'un utilisateur a été créé comme suit.

```
CREATE USER script_user IDENTIFIED BY password;
```

Pour une base de données sur site, définissez les autorisations minimales telles qu'indiquées ci-dessous pour *script_user*.

```
GRANT CREATE SESSION TO script_user;
GRANT SELECT on V$DATABASE to script_user;
GRANT SELECT on V$VERSION to script_user;
GRANT SELECT on GV$SGA to script_user;
GRANT SELECT on GV$INSTANCE to script_user;
GRANT SELECT on GV$DATAGUARD_CONFIG to script_user;
GRANT SELECT on GV$LOG to script_user;
GRANT SELECT on DBA_TABLESPACES to script_user;
GRANT SELECT on DBA_DATA_FILES to script_user;
GRANT SELECT on DBA_SEGMENTS to script_user;
GRANT SELECT on DBA_LOBS to script_user;
GRANT SELECT on V$ARCHIVED_LOG to script_user;
GRANT SELECT on DBA_TAB_MODIFICATIONS to script_user;
GRANT SELECT on DBA_TABLES to script_user;
GRANT SELECT on DBA_TAB_PARTITIONS to script_user;
GRANT SELECT on DBA_MVIEWS to script_user;
GRANT SELECT on DBA_OBJECTS to script_user;
GRANT SELECT on DBA_TAB_COLUMNS to script_user;
GRANT SELECT on DBA_LOG_GROUPS to script_user;
GRANT SELECT on DBA_LOG_GROUP_COLUMNS to script_user;
GRANT SELECT on V$ARCHIVE_DEST to script_user;
GRANT SELECT on DBA_SYS_PRIVS to script_user;
GRANT SELECT on DBA_TAB_PRIVS to script_user;
```

```
GRANT SELECT on DBA_TYPES to script_user;
GRANT SELECT on DBA_CONSTRAINTS to script_user;
GRANT SELECT on V$TRANSACTION to script_user;
GRANT SELECT on GV$ASM_DISK_STAT to script_user;
GRANT SELECT on GV$SESSION to script_user;
GRANT SELECT on GV$SQL to script_user;
GRANT SELECT on DBA_ENCRYPTED_COLUMNS to script_user;
GRANT SELECT on DBA_PDBS to script_user;

GRANT EXECUTE on dbms_utility to script_user;
```

Pour une base de données Amazon RDS, définissez les autorisations minimales telles qu'indiquées ci-dessous.

```
GRANT CREATE SESSION TO script_user;
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$DATABASE','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$VERSION','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('GV_$SGA','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('GV_$INSTANCE','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('GV_
$DATAGUARD_CONFIG','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('GV_$LOG','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TABLESPACES','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_DATA_FILES','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_SEGMENTS','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_LOBS','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$ARCHIVED_LOG','script_user','SELECT');
exec
 rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_MODIFICATIONS','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TABLES','script_user','SELECT');
exec
 rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_PARTITIONS','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_MVIEWS','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_OBJECTS','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_COLUMNS','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_LOG_GROUPS','script_user','SELECT');
exec
 rdsadmin.rdsadmin_util.grant_sys_object('DBA_LOG_GROUP_COLUMNS','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$ARCHIVE_DEST','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_SYS_PRIVS','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_PRIVS','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TYPES','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_CONSTRAINTS','script_user','SELECT');
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$TRANSACTION','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('GV_
$ASM_DISK_STAT','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('GV_$SESSION','script_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('GV_$SQL','script_user','SELECT');
exec
 rdsadmin.rdsadmin_util.grant_sys_object('DBA_ENCRYPTED_COLUMNS','script_user','SELECT');

exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_PDBS','script_user','SELECT');

exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_UTILITY','script_user','EXECUTE');
```

Vous trouverez ci-dessous des descriptions indiquant comment télécharger, passer en revue et exécuter chaque script d'assistance SQL*Plus disponible pour Oracle. Vous trouverez également comment passer en revue et charger la sortie du script dans votre dossier AWS Support.

Rubriques

- Script awsdms_support_collector_oracle.sql

Script awsdms_support_collector_oracle.sql

Téléchargez le script awsdms_support_collector_oracle.sql.

Ce script collecte des informations sur la configuration de la base de données Oracle. N'oubliez pas de vérifier la somme de contrôle sur le script et, si cette somme de contrôle est correcte, passez en revue le code SQL du script pour mettre en commentaires toute partie du code que vous n'envisagez pas sereinement d'exécuter. Une fois que vous êtes satisfait de l'intégrité et du contenu du script, vous pouvez exécuter ce dernier.

Pour exécuter le script et charger les résultats dans votre dossier de support

1. Exécutez le script à partir de votre environnement de base de données à l'aide de la ligne de commande SQL*Plus suivante.

   ```
   SQL> @awsdms_support_collector_oracle.sql
   ```

   <result>

   Le script affiche une brève description et vous invite à poursuivre ou à abandonner l'exécution. Appuyez sur [Entrée] pour continuer.

</result>

2. À l'invite suivante, entrez le nom d'un seul des schémas que vous souhaitez migrer.
3. À l'invite suivante, entrez le nom de l'utilisateur (*script_user*) que vous avez défini pour vous connecter à la base de données.
4. À l'invite suivante, entrez le nombre de jours de données que vous souhaitez examiner ou acceptez la valeur par défaut. Le script collecte ensuite les données spécifiées à partir de la base de données.
   <result>

   Une fois le script terminé, il affiche le nom du fichier HTML de sortie, par exemple `dms_support_oracle-2020-06-22-13-20-39-ORCL.html`. Le script enregistre ce fichier dans votre répertoire de travail.
   </result>
5. Passez en revue ce fichier HTML et supprimez toutes les informations que vous ne souhaitez pas partager. Lorsque le fichier HTML vous semble approprié à partager, chargez-le dans votre dossier AWS Support. Pour plus d'informations sur le chargement de ce fichier, consultez Utilisation de scripts d'assistance au diagnostic dans AWS DMS.

Scripts d'assistance au diagnostic SQL Server

Vous trouverez ci-dessous une description des scripts d'assistance au diagnostic disponibles pour analyser une base de données sur site ou Amazon RDS for SQL Server dans votre configuration de migration AWS DMS. Ces scripts fonctionnent avec un point de terminaison source ou cible. Pour une base de données sur site, exécutez ces scripts dans l'utilitaire de ligne de commande sqlcmd. Pour plus d'informations sur l'utilisation de cet utilitaire, consultez sqlcmd : utiliser l'utilitaire dans la documentation Microsoft.

Pour une base de données Amazon RDS, vous ne pouvez pas vous connecter à l'aide de l'utilitaire de ligne de commande sqlcmd. Exécutez plutôt ces scripts à l'aide d'un outil client quelconque connecté à Amazon RDS SQL Server.

Avant d'exécuter le script, veillez à ce que le compte d'utilisateur que vous utilisez dispose des autorisations nécessaires pour accéder à la base de données SQL Server. Pour une base de données sur site et une base de données Amazon RDS, vous pouvez utiliser les mêmes autorisations que celles que vous utilisez pour accéder à la base de données SQL Server sans le rôle `SysAdmin`.

Rubriques

- Configuration des autorisations minimales pour une base de données SQL Server sur site
- Configuration des autorisations minimales pour une base de données Amazon RDS SQL Server
- Configuration de la réplication continue sur une instance SQL Server autonome : sans le rôle sysadmin
- Configuration de la réplication continue sur une instance SQL Server dans un environnement de groupe de disponibilité : sans le rôle sysadmin
- Scripts d'assistance SQL Server

Configuration des autorisations minimales pour une base de données SQL Server sur site

Pour configurer les autorisations minimales d'exécution pour une base de données SQL Server sur site

1. Créez un nouveau compte SQL Server avec une authentification par mot de passe utilisant SQL Server Management Studio (SSMS), par exemple *on-prem-user*.
2. Dans la section Mappages d'utilisateurs de SSMS, choisissez les bases de données MSDB et MASTER (ce qui confère une autorisation publique) et affectez le rôle DB_OWNER à la base de données dans laquelle vous voulez exécuter le script.
3. Ouvrez le menu contextuel (clic droit) pour le nouveau compte et choisissez Sécurité pour accorder explicitement le privilège Connect SQL.
4. Exécutez les commandes d'octroi suivantes.

```
GRANT VIEW SERVER STATE TO on-prem-user;
USE MSDB;
GRANT SELECT ON MSDB.DBO.BACKUPSET TO on-prem-user;
GRANT SELECT ON MSDB.DBO.BACKUPMEDIAFAMILY TO on-prem-user;
GRANT SELECT ON MSDB.DBO.BACKUPFILE TO on-prem-user;
```

Configuration des autorisations minimales pour une base de données Amazon RDS SQL Server

Pour exécuter une base de données Amazon RDS SQL Server avec les autorisations minimales

1. Créez un nouveau compte SQL Server avec une authentification par mot de passe utilisant SQL Server Management Studio (SSMS), par exemple *rds-user*.
2. Dans la section Mappages d'utilisateurs de SSMS, choisissez la base de données MSDB (ce qui confère une autorisation publique) et affectez le rôle DB_OWNER à la base de données dans laquelle vous voulez exécuter le script.
3. Ouvrez le menu contextuel (clic droit) pour le nouveau compte et choisissez Sécurité pour accorder explicitement le privilège Connect SQL.
4. Exécutez les commandes d'octroi suivantes.

```
GRANT VIEW SERVER STATE TO rds-user;
USE MSDB;
GRANT SELECT ON MSDB.DBO.BACKUPSET TO rds-user;
GRANT SELECT ON MSDB.DBO.BACKUPMEDIAFAMILY TO rds-user;
GRANT SELECT ON MSDB.DBO.BACKUPFILE TO rds-user;
```

Configuration de la réplication continue sur une instance SQL Server autonome : sans le rôle sysadmin

Cette section décrit comment configurer la réplication continue pour une source de base de données SQL Server autonome qui n'exige pas que le compte d'utilisateur dispose des privilèges sysadmin.

> **Note**
>
> Après avoir exécuté les étapes de cette section, l'utilisateur DMS qui n'est pas un administrateur système sera autorisé à effectuer les étapes suivantes :
>
> - Lire les modifications du journal de transactions en ligne
> - Accéder au disque pour lire les modifications des fichiers de sauvegarde des journaux de transactions
> - Ajouter ou modifier la publication utilisée par DMS
> - Ajouter des articles à la publication

1. Configurez Microsoft SQL Server pour la réplication comme décrit dans Capture des modifications de données pour SQL Server autogéré sur site ou sur Amazon EC2.
2. Activez la réplication Microsoft sur la base de données source. Cela peut être fait manuellement ou en exécutant la tâche une fois en tant qu'utilisateur sysadmin.
3. Créez le schéma `awsdms` sur la base de données source à l'aide du script suivant :

```
use master
go
create schema awsdms
go


-- Create the table valued function [awsdms].[split_partition_list] on the Master
 database, as follows:
USE [master]
GO

set ansi_nulls on
go

set quoted_identifier on
go

if (object_id('[awsdms].[split_partition_list]','TF')) is not null

drop function [awsdms].[split_partition_list];

go

create function [awsdms].[split_partition_list]

(

@plist varchar(8000), –A delimited list of partitions

@dlm nvarchar(1) –Delimiting character

)

returns @partitionsTable table –Table holding the BIGINT values of the string
 fragments
```

```
(

pid bigint primary key

)

as

begin

declare @partition_id bigint;

declare @dlm_pos integer;

declare @dlm_len integer;

set @dlm_len = len(@dlm);

while (charindex(@dlm,@plist)>0)

begin

set @dlm_pos = charindex(@dlm,@plist);

set @partition_id = cast( ltrim(rtrim(substring(@plist,1,@dlm_pos-1))) as bigint);

insert into @partitionsTable (pid) values (@partition_id)

set @plist = substring(@plist,@dlm_pos+@dlm_len,len(@plist));

end

set @partition_id = cast (ltrim(rtrim(@plist)) as bigint);

insert into @partitionsTable (pid) values ( @partition_id );

return

end

GO
```

4. Créez la procédure `[awsdms].[rtm_dump_dblog]` sur la base de données Master à l'aide du script suivant :

```
use [MASTER]

go

if (object_id('[awsdms].[rtm_dump_dblog]','P')) is not null drop procedure
 [awsdms].[rtm_dump_dblog];
go


set ansi_nulls on
go

set quoted_identifier on
GO



CREATE procedure [awsdms].[rtm_dump_dblog]

(

@start_lsn varchar(32),

@seqno integer,

@filename varchar(260),

@partition_list varchar(8000), — A comma delimited list: P1,P2,... Pn

@programmed_filtering integer,

@minPartition bigint,

@maxPartition bigint

)

as begin

declare @start_lsn_cmp varchar(32); — Stands against the GT comparator

SET NOCOUNT ON — - Disable "rows affected display"
```

```
set @start_lsn_cmp = @start_lsn;

if (@start_lsn_cmp) is null

set @start_lsn_cmp = '00000000:00000000:0000';

if (@partition_list is null)

begin

RAISERROR ('Null partition list waspassed',16,1);

return

end

if (@start_lsn) is not null

set @start_lsn = '0x'+@start_lsn;

if (@programmed_filtering=0)



SELECT

[Current LSN],

[operation],

[Context],

[Transaction ID],

[Transaction Name],

[Begin Time],

[End Time],

[Flag Bits],

[PartitionID],
```

```
[Page ID],

[Slot ID],

[RowLog Contents 0],

[Log Record],

[RowLog Contents 1]

FROM

fn_dump_dblog (

@start_lsn, NULL, N'DISK', @seqno, @filename,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default)

where [Current LSN] collate SQL_Latin1_General_CP1_CI_AS > @start_lsn_cmp collate
 SQL_Latin1_General_CP1_CI_AS

and

(

( [operation] in ('LOP_BEGIN_XACT','LOP_COMMIT_XACT','LOP_ABORT_XACT') )

or
```

```
( [operation] in ('LOP_INSERT_ROWS','LOP_DELETE_ROWS','LOP_MODIFY_ROW')

and

( ( [context] in ('LCX_HEAP','LCX_CLUSTERED','LCX_MARK_AS_GHOST') ) or ([context] =
 'LCX_TEXT_MIX' and (datalength([RowLog Contents 0]) in (0,1))))

and [PartitionID] in ( select * from master.awsdms.split_partition_list
 (@partition_list,','))

)

or

([operation] = 'LOP_HOBT_DDL')

)


else


SELECT

[Current LSN],

[operation],

[Context],

[Transaction ID],

[Transaction Name],

[Begin Time],

[End Time],

[Flag Bits],

[PartitionID],

[Page ID],
```

```
[Slot ID],

[RowLog Contents 0],

[Log Record],

[RowLog Contents 1] – After Image

FROM

fn_dump_dblog (

@start_lsn, NULL, N'DISK', @seqno, @filename,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default)

where [Current LSN] collate SQL_Latin1_General_CP1_CI_AS > @start_lsn_cmp collate
 SQL_Latin1_General_CP1_CI_AS

and

(

( [operation] in ('LOP_BEGIN_XACT','LOP_COMMIT_XACT','LOP_ABORT_XACT') )

or
```

```
( [operation] in ('LOP_INSERT_ROWS','LOP_DELETE_ROWS','LOP_MODIFY_ROW')

and

( ( [context] in ('LCX_HEAP','LCX_CLUSTERED','LCX_MARK_AS_GHOST') ) or ([context] =
 'LCX_TEXT_MIX' and (datalength([RowLog Contents 0]) in (0,1))))

and ([PartitionID] is not null) and ([PartitionID] >= @minPartition and
 [PartitionID]<=@maxPartition)

)

or

([operation] = 'LOP_HOBT_DDL')

)



SET NOCOUNT OFF – Re-enable "rows affected display"

end

GO
```

5. Créez le certificat sur la base de données Master à l'aide du script suivant :

```
Use [master]
Go

CREATE CERTIFICATE [awsdms_rtm_dump_dblog_cert] ENCRYPTION BY PASSWORD =
 N'@5trongpassword'

WITH SUBJECT = N'Certificate for FN_DUMP_DBLOG Permissions';
```

6. Créez la connexion à partir du certificat à l'aide du script suivant :

```
Use [master]
Go

CREATE LOGIN awsdms_rtm_dump_dblog_login FROM CERTIFICATE
 [awsdms_rtm_dump_dblog_cert];
```

7. Ajoutez la connexion au rôle serveur sysadmin à l'aide du script suivant :

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_dump_dblog_login];
```

8. Ajoutez la signature à [master].[awsdms].[rtm_dump_dblog] utilisant le certificat, à l'aide du script suivant :

```
Use [master]
GO
ADD SIGNATURE
TO [master].[awsdms].[rtm_dump_dblog] BY CERTIFICATE [awsdms_rtm_dump_dblog_cert]
 WITH PASSWORD = '@5trongpassword';
```

> **Note**
>
> Si vous recréez la procédure stockée, vous devez ajouter à nouveau la signature.

9. Créez [awsdms]. [rtm_position_1st_timestamp] dans la base de données principale à l'aide du script suivant :

```
use [master]
    if object_id('[awsdms].[rtm_position_1st_timestamp]','P') is not null
    DROP PROCEDURE [awsdms].[rtm_position_1st_timestamp];
    go
    create procedure [awsdms].[rtm_position_1st_timestamp]
    (
    @dbname                sysname,      -- Database name
    @seqno                 integer,      -- Backup set sequence/position number
 within file
    @filename              varchar(260), -- The backup filename
    @1stTimeStamp          varchar(40)   -- The timestamp to position by
    )
    as begin

    SET NOCOUNT ON       -- Disable "rows affected display"

    declare @firstMatching table
    (
    cLsn varchar(32),
    bTim datetime
    )
```

```
   declare @sql nvarchar(4000)
   declare @nl                      char(2)
   declare @tb                      char(2)
   declare @fnameVar                nvarchar(254) = 'NULL'

   set @nl  = char(10); -- New line
   set @tb  = char(9)   -- Tab separator

   if (@filename is not null)
   set @fnameVar = ''''+@filename +''''

   set @sql='use ['+@dbname+'];'+@nl+
   'select top 1 [Current LSN],[Begin Time]'+@nl+
   'FROM fn_dump_dblog (NULL, NULL, NULL, '+ cast(@seqno as varchar(10))+','+
@fnameVar+','+@nl+
   @tb+'default, default, default, default, default, default, default,'+@nl+
   @tb+'default, default, default, default, default, default, default,'+@nl+
   @tb+'default, default, default, default, default, default, default,'+@nl+
   @tb+'default, default, default, default, default, default, default,'+@nl+
   @tb+'default, default, default, default, default, default, default,'+@nl+
   @tb+'default, default, default, default, default, default, default,'+@nl+
   @tb+'default, default, default, default, default, default, default,'+@nl+
   @tb+'default, default, default, default, default, default, default,'+@nl+
   @tb+'default, default, default, default, default, default, default)'+@nl+
   'where operation=''LOP_BEGIN_XACT''' +@nl+
   'and [Begin Time]>= cast('+''''+@1stTimeStamp+''''+' as datetime)'+@nl

   --print @sql
   delete from  @firstMatching
   insert into @firstMatching  exec sp_executesql @sql    -- Get them all

   select top 1 cLsn as [matching LSN],convert(varchar,bTim,121) as [matching
Timestamp] from @firstMatching;

   SET NOCOUNT OFF      -- Re-enable "rows affected display"

   end
   GO
```

10. Créez le certificat sur la base de données Master à l'aide du script suivant :

```
Use [master]
Go
```

```
CREATE CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
ENCRYPTION BY PASSWORD = '@5trongpassword'
WITH SUBJECT = N'Certificate for FN_POSITION_1st_TIMESTAMP Permissions';
```

11. Créez la connexion à partir du certificat à l'aide du script suivant :

```
Use [master]
Go
CREATE LOGIN awsdms_rtm_position_1st_timestamp_login FROM CERTIFICATE
 [awsdms_rtm_position_1st_timestamp_cert];
```

12. Ajoutez la connexion au rôle sysadmin à l'aide du script suivant :

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_position_1st_timestamp_login];
```

13. Ajoutez la signature à [master].[awsdms].[rtm_position_1st_timestamp] en utilisant le certificat à l'aide du script suivant :

```
Use [master]
    GO
    ADD SIGNATURE
    TO [master].[awsdms].[rtm_position_1st_timestamp]
    BY CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
    WITH PASSWORD = '@5trongpassword';
```

14. Accordez à l'utilisateur DMS l'accès lui permettant d'exécuter la nouvelle procédure stockée à l'aide du script suivant :

```
use master
go
GRANT execute on [awsdms].[rtm_position_1st_timestamp] to dms_user;
```

15. Créez un utilisateur avec les autorisations et les rôles suivants dans chacune des bases de données suivantes :

> **Note**
>
> Vous devez créer le compte d'utilisateur dmsnosysadmin avec le même SID sur chaque réplica. La requête SQL suivante peut aider à vérifier la valeur SID du compte dmsnosysadmin sur chaque réplica. Pour plus d'informations sur la création d'un utilisateur, consultez CREATE USER (Transact-SQL) dans la documentation sur

Microsoft SQL Server. Pour plus d'informations sur la création de comptes d'utilisateur SQL pour la base de données Azure SQL, consultez Géoréplication active.

```
use master
go
grant select on sys.fn_dblog to [DMS_user]
grant view any definition to [DMS_user]
grant view server state to [DMS_user]—(should be granted to the login).
grant execute on sp_repldone to [DMS_user]
grant execute on sp_replincrementlsn to [DMS_user]
grant execute on sp_addpublication to [DMS_user]
grant execute on sp_addarticle to [DMS_user]
grant execute on sp_articlefilter to [DMS_user]
grant select on [awsdms].[split_partition_list] to [DMS_user]
grant execute on [awsdms].[rtm_dump_dblog] to [DMS_user]
```

```
use MSDB
go
grant select on msdb.dbo.backupset to [DMS_user]
grant select on msdb.dbo.backupmediafamily to [DMS_user]
grant select on msdb.dbo.backupfile to [DMS_user]
```

Exécutez le script suivant sur la base de données source :

```
EXEC sp_addrolemember N'db_owner', N'DMS_user'
use Source_DB
go
```

16. Enfin, ajoutez un attribut de connexion supplémentaire (ECA) au point de terminaison SQL Server source :

```
enableNonSysadminWrapper=true;
```

Configuration de la réplication continue sur une instance SQL Server dans un environnement de groupe de disponibilité : sans le rôle sysadmin

Cette section décrit comment configurer la réplication continue pour une source de base de données SQL Server dans un environnement de groupe de disponibilité qui n'exige pas que le compte d'utilisateur dispose des privilèges sysadmin.

> **Note**
>
> Après avoir exécuté les étapes de cette section, l'utilisateur DMS qui n'est pas un administrateur système sera autorisé à effectuer les étapes suivantes :
>
> - Lire les modifications du journal de transactions en ligne
> - Accéder au disque pour lire les modifications des fichiers de sauvegarde des journaux de transactions
> - Ajouter ou modifier la publication utilisée par DMS
> - Ajouter des articles à la publication

Pour configurer la réplication continue sans utiliser l'utilisateur sysadmin dans un environnement de groupe de disponibilité

1. Configurez Microsoft SQL Server pour la réplication comme décrit dans Capture des modifications de données pour SQL Server autogéré sur site ou sur Amazon EC2.
2. Activez la réplication Microsoft sur la base de données source. Cela peut être fait manuellement ou en exécutant la tâche une fois en ayant recours à un utilisateur sysadmin.

   > **Note**
   >
   > Vous devez configurer le distributeur de réplication Microsoft comme local ou de manière à autoriser l'accès aux utilisateurs non sysadmin via le serveur lié associé.

3. Si l'option de point de terminaison Utiliser exclusivement sp_repldone au sein d'une seule tâche est activée, arrêtez la tâche Log Reader de réplication Microsoft.
4. Sur chaque réplica, procédez comme suit :

   1. Créez le schéma `[awsdms]`[awsdms] dans la base de données Master :

```
CREATE SCHEMA [awsdms]
```

2. Créez la fonction à valeur de table `[awsdms].[split_partition_list]` sur la base de données Master :

```
USE [master]
GO

SET ansi_nulls on
GO

SET quoted_identifier on
GO

IF (object_id('[awsdms].[split_partition_list]','TF')) is not null
  DROP FUNCTION [awsdms].[split_partition_list];
GO

CREATE FUNCTION [awsdms].[split_partition_list]
(
  @plist varchar(8000),    --A delimited list of partitions
  @dlm nvarchar(1)    --Delimiting character
)
RETURNS @partitionsTable table --Table holding the BIGINT values of the string
 fragments
(
  pid bigint primary key
)
AS
BEGIN
  DECLARE @partition_id bigint;
  DECLARE @dlm_pos integer;
  DECLARE @dlm_len integer;
  SET @dlm_len = len(@dlm);
  WHILE (charindex(@dlm,@plist)>0)
  BEGIN
    SET @dlm_pos = charindex(@dlm,@plist);
    SET @partition_id = cast( ltrim(rtrim(substring(@plist,1,@dlm_pos-1))) as
 bigint);
    INSERT into @partitionsTable (pid) values (@partition_id)
    SET @plist = substring(@plist,@dlm_pos+@dlm_len,len(@plist));
  END
```

```
  SET @partition_id = cast (ltrim(rtrim(@plist)) as bigint);
  INSERT into @partitionsTable (pid) values (  @partition_id  );
  RETURN
END
GO
```

3. Créez la procédure `[awsdms].[rtm_dump_dblog]` sur la base de données Master :

```
USE [MASTER]
GO

IF (object_id('[awsdms].[rtm_dump_dblog]','P')) is not null
  DROP PROCEDURE [awsdms].[rtm_dump_dblog];
GO

SET ansi_nulls on
GO

SET quoted_identifier on
GO

CREATE PROCEDURE [awsdms].[rtm_dump_dblog]
(
  @start_lsn            varchar(32),
  @seqno                integer,
  @filename             varchar(260),
  @partition_list       varchar(8000), -- A comma delimited list: P1,P2,... Pn
  @programmed_filtering integer,
  @minPartition         bigint,
  @maxPartition         bigint
)
AS
BEGIN

  DECLARE @start_lsn_cmp varchar(32); -- Stands against the GT comparator

  SET NOCOUNT ON  -- Disable "rows affected display"

  SET @start_lsn_cmp = @start_lsn;
  IF (@start_lsn_cmp) is null
    SET @start_lsn_cmp = '00000000:00000000:0000';

  IF (@partition_list is null)
    BEGIN
```

```
      RAISERROR ('Null partition list was passed',16,1);
      return
      --set @partition_list = '0,';    -- A dummy which is never matched
    END

 IF (@start_lsn) is not null
    SET @start_lsn = '0x'+@start_lsn;

 IF (@programmed_filtering=0)
    SELECT
      [Current LSN],
      [operation],
      [Context],
      [Transaction ID],
      [Transaction Name],
      [Begin Time],
      [End Time],
      [Flag Bits],
      [PartitionID],
      [Page ID],
      [Slot ID],
      [RowLog Contents 0],
      [Log Record],
      [RowLog Contents 1] -- After Image
    FROM
      fn_dump_dblog (
        @start_lsn, NULL, N'DISK', @seqno, @filename,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default)
    WHERE
      [Current LSN] collate SQL_Latin1_General_CP1_CI_AS > @start_lsn_cmp collate
SQL_Latin1_General_CP1_CI_AS -- This aims for implementing FN_DBLOG based on GT
comparator.
      AND
      (
        (  [operation] in ('LOP_BEGIN_XACT','LOP_COMMIT_XACT','LOP_ABORT_XACT') )
        OR
```

```
       (  [operation] in ('LOP_INSERT_ROWS','LOP_DELETE_ROWS','LOP_MODIFY_ROW')
         AND
         ( ( [context]   in ('LCX_HEAP','LCX_CLUSTERED','LCX_MARK_AS_GHOST') )
or ([context] = 'LCX_TEXT_MIX') )
         AND
         [PartitionID] in ( select * from master.awsdms.split_partition_list
(@partition_list,','))
       )
     OR
     ([operation] = 'LOP_HOBT_DDL')
   )
   ELSE
     SELECT
       [Current LSN],
       [operation],
       [Context],
       [Transaction ID],
       [Transaction Name],
       [Begin Time],
       [End Time],
       [Flag Bits],
       [PartitionID],
       [Page ID],
       [Slot ID],
       [RowLog Contents 0],
       [Log Record],
       [RowLog Contents 1] -- After Image
     FROM
       fn_dump_dblog (
         @start_lsn, NULL, N'DISK', @seqno, @filename,
         default, default, default, default, default, default, default,
         default, default, default, default, default, default, default,
         default, default, default, default, default, default, default,
         default, default, default, default, default, default, default,
         default, default, default, default, default, default, default,
         default, default, default, default, default, default, default,
         default, default, default, default, default, default, default,
         default, default, default, default, default, default, default)
     WHERE [Current LSN] collate SQL_Latin1_General_CP1_CI_AS > @start_lsn_cmp
collate SQL_Latin1_General_CP1_CI_AS -- This aims for implementing FN_DBLOG
based on GT comparator.
     AND
     (
```

```
        (  [operation] in ('LOP_BEGIN_XACT','LOP_COMMIT_XACT','LOP_ABORT_XACT') )
        OR
        (  [operation] in ('LOP_INSERT_ROWS','LOP_DELETE_ROWS','LOP_MODIFY_ROW')
          AND
          ( ( [context]   in ('LCX_HEAP','LCX_CLUSTERED','LCX_MARK_AS_GHOST') )
 or ([context] = 'LCX_TEXT_MIX') )
          AND ([PartitionID] is not null) and ([PartitionID] >= @minPartition and
 [PartitionID]<=@maxPartition)
        )
        OR
        ([operation] = 'LOP_HOBT_DDL')
      )
      SET NOCOUNT OFF -- Re-enable "rows affected display"
END
GO
```

4. Créez un certificat sur la base de données Master :

```
USE [master]
GO
CREATE CERTIFICATE [awsdms_rtm_dump_dblog_cert]
  ENCRYPTION BY PASSWORD = N'@hardpassword1'
  WITH SUBJECT = N'Certificate for FN_DUMP_DBLOG Permissions'
```

5. Créez une connexion à partir du certificat :

```
USE [master]
GO
CREATE LOGIN awsdms_rtm_dump_dblog_login FROM CERTIFICATE
  [awsdms_rtm_dump_dblog_cert];
```

6. Ajoutez la connexion au rôle serveur sysadmin :

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_dump_dblog_login];
```

7. Ajoutez la signature à la procédure [master].[awsdms].[rtm_dump_dblog] utilisant le certificat :

```
USE [master]
GO

ADD SIGNATURE
  TO [master].[awsdms].[rtm_dump_dblog]
  BY CERTIFICATE [awsdms_rtm_dump_dblog_cert]
```

```
  WITH PASSWORD = '@hardpassword1';
```

> **Note**
>
> Si vous recréez la procédure stockée, vous devez ajouter à nouveau la signature.

8. Créez la procédure `[awsdms].[rtm_position_1st_timestamp]` sur la base de données Master :

```
USE [master]
IF object_id('[awsdms].[rtm_position_1st_timestamp]','P') is not null
  DROP PROCEDURE [awsdms].[rtm_position_1st_timestamp];
GO
CREATE PROCEDURE [awsdms].[rtm_position_1st_timestamp]
(
  @dbname                sysname,      -- Database name
  @seqno                 integer,      -- Backup set sequence/position number
 within file
  @filename              varchar(260), -- The backup filename
  @1stTimeStamp          varchar(40)   -- The timestamp to position by
)
AS
BEGIN
  SET NOCOUNT ON       -- Disable "rows affected display"

  DECLARE @firstMatching table
  (
    cLsn varchar(32),
    bTim datetime
  )
  DECLARE @sql nvarchar(4000)
  DECLARE @nl                        char(2)
  DECLARE @tb                        char(2)
  DECLARE @fnameVar                  sysname = 'NULL'

  SET @nl  = char(10); -- New line
  SET @tb  = char(9)   -- Tab separator

  IF (@filename is not null)
    SET @fnameVar = ''''+@filename +''''
  SET @filename = ''''+@filename +''''
  SET @sql='use ['+@dbname+'];'+@nl+
```

```
    'SELECT TOP 1 [Current LSN],[Begin Time]'+@nl+
    'FROM fn_dump_dblog (NULL, NULL, NULL, '+ cast(@seqno as varchar(10))+','+
 @filename +','+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default)'+@nl+
    'WHERE operation=''LOP_BEGIN_XACT''' +@nl+
    'AND [Begin Time]>= cast('+''''+@1stTimeStamp+''''+' as datetime)'+@nl

    --print @sql
    DELETE FROM @firstMatching
    INSERT INTO @firstMatching  exec sp_executesql @sql    -- Get them all
    SELECT TOP 1 cLsn as [matching LSN],convert(varchar,bTim,121) AS[matching
 Timestamp] FROM @firstMatching;

    SET NOCOUNT OFF      -- Re-enable "rows affected display"

END
GO
```

9. Créez un certificat sur la base de données Master :

```
USE [master]
GO
CREATE CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
  ENCRYPTION BY PASSWORD = N'@hardpassword1'
  WITH SUBJECT = N'Certificate for FN_POSITION_1st_TIMESTAMP Permissions';
```

10. Créez une connexion à partir du certificat :

```
USE [master]
GO
CREATE LOGIN awsdms_rtm_position_1st_timestamp_login FROM CERTIFICATE
  [awsdms_rtm_position_1st_timestamp_cert];
```

11. Ajoutez la connexion au rôle serveur sysadmin :

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER
 [awsdms_rtm_position_1st_timestamp_login];
```

12 Ajoutez la signature à la procédure `[master].[awsdms].[rtm_position_1st_timestamp]` utilisant le certificat :

```
USE [master]
GO
ADD SIGNATURE
  TO [master].[awsdms].[rtm_position_1st_timestamp]
  BY CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
  WITH PASSWORD = '@hardpassword1';
```

> **Note**
>
> Si vous recréez la procédure stockée, vous devez ajouter à nouveau la signature.

13 Créez un utilisateur avec les autorisations/rôles suivants dans chacune des bases de données suivantes :

> **Note**
>
> Vous devez créer le compte d'utilisateur dmsnosysadmin avec le même SID sur chaque réplica. La requête SQL suivante peut aider à vérifier la valeur SID du compte dmsnosysadmin sur chaque réplica. Pour plus d'informations sur la création d'un utilisateur, consultez CREATE USER (Transact-SQL) dans la documentation sur Microsoft SQL Server. Pour plus d'informations sur la création de comptes d'utilisateur SQL pour la base de données Azure SQL, consultez Géoréplication active.

```
SELECT @@servername servername, name, sid, create_date, modify_date
  FROM sys.server_principals
  WHERE name = 'dmsnosysadmin';
```

14 Accordez des autorisations sur la base de données Master pour chaque réplica :

```
USE master
GO
```

```
GRANT select on sys.fn_dblog to dmsnosysadmin;
GRANT view any definition to dmsnosysadmin;
GRANT view server state to dmsnosysadmin -- (should be granted to the login).
GRANT execute on sp_repldone to dmsnosysadmin;
GRANT execute on sp_replincrementlsn to dmsnosysadmin;
GRANT execute on sp_addpublication to dmsnosysadmin;
GRANT execute on sp_addarticle to dmsnosysadmin;
GRANT execute on sp_articlefilter to dmsnosysadmin;
GRANT select on [awsdms].[split_partition_list] to dmsnosysadmin;
GRANT execute on [awsdms].[rtm_dump_dblog] to dmsnosysadmin;
GRANT execute on [awsdms].[rtm_position_1st_timestamp] to dmsnosysadmin;
```

15 Accordez des autorisations sur la base de données msdb pour chaque réplica :

```
USE msdb
GO
GRANT select on msdb.dbo.backupset to dmsnosysadmin
GRANT select on msdb.dbo.backupmediafamily to dmsnosysadmin
GRANT select on msdb.dbo.backupfile to dmsnosysadmin
```

16 Ajoutez le rôle `db_owner` à `dmsnosysadmin` sur la base de données source. La base de données étant synchronisée, vous pouvez vous contenter d'ajouter le rôle sur le réplica principal uniquement.

```
use <source DB>
GO
EXEC sp_addrolemember N'db_owner', N'dmsnosysadmin'
```

Scripts d'assistance SQL Server

Les rubriques suivantes décrivent comment télécharger, passer en revue et exécuter chaque script d'assistance disponible pour SQL Server. Elles décrivent également comment passer en revue et charger le résultat du script dans votre dossier AWS Support.

Rubriques

- Script awsdms_support_collector_sql_server.sql

Script awsdms_support_collector_sql_server.sql

Téléchargez le script awsdms_support_collector_sql_server.sql.

> **Note**
>
> Exécutez ce script d'assistance au diagnostic SQL Server sur SQL Server 2014 et versions ultérieures uniquement.

Ce script collecte des informations sur la configuration de la base de données SQL Server. N'oubliez pas de vérifier la somme de contrôle sur le script et, si cette somme de contrôle est correcte, passez en revue le code SQL du script pour mettre en commentaires toute partie du code que vous n'envisagez pas sereinement d'exécuter. Une fois que vous êtes satisfait de l'intégrité et du contenu du script, vous pouvez exécuter ce dernier.

Pour exécuter le script pour une base de données SQL Server sur site

1. Exécutez le script à l'aide de la ligne de commande sqlcmd suivante.

```
sqlcmd -Uon-prem-user -Ppassword -SDMS-SQL17AG-N1 -y 0
-iC:\Users\admin\awsdms_support_collector_sql_server.sql -oC:\Users\admin
\DMS_Support_Report_SQLServer.html -dsqlserverdb01
```

   Les paramètres spécifiés de la commande sqlcmd incluent les suivants :

   - `-U` : nom d'utilisateur de la base de données.
   - `-P` : mot de passe d'utilisateur de la base de données.
   - `-S` : nom du serveur de base de données SQL Server.
   - `-y` : largeur maximale des colonnes de sortie de l'utilitaire sqlcmd. La valeur 0 indique que les colonnes ont une largeur illimitée.
   - `-i` : chemin du script d'assistance à exécuter, dans ce cas, `awsdms_support_collector_sql_server.sql`.
   - `-o` : chemin du fichier HTML de sortie, avec un nom de fichier que vous spécifiez, contenant les informations de configuration de base de données collectées.
   - `-d` : nom de la base de données SQL Server.

2. Une fois le script terminé, passez en revue le fichier HTML de sortie et supprimez toutes les informations que vous ne souhaitez pas partager. Lorsque le fichier HTML vous semble approprié à partager, chargez-le dans votre dossier AWS Support. Pour plus d'informations sur le chargement de ce fichier, consultez Utilisation de scripts d'assistance au diagnostic dans AWS DMS.

Avec Amazon RDS for SQL Server, vous ne pouvez pas vous connecter à l'aide de l'utilitaire de ligne de commande sqlcmd. Procédez donc comme suit.

Pour exécuter le script pour une base de données RDS SQL Server

1. Exécutez le script à l'aide d'un outil client quelconque vous permettant de vous connecter à RDS SQL Server en tant qu'utilisateur `Master` et d'enregistrer la sortie en tant que fichier HTML.
2. Passez en revue le fichier HTML de sortie et supprimez toutes les informations que vous ne souhaitez pas partager. Lorsque le fichier HTML vous semble approprié à partager, chargez-le dans votre dossier AWS Support. Pour plus d'informations sur le chargement de ce fichier, consultez Utilisation de scripts d'assistance au diagnostic dans AWS DMS.

Scripts d'assistance au diagnostic pour les bases de données compatibles MySQL

Vous trouverez ci-dessous les scripts d'assistance au diagnostic disponibles pour analyser une base de données sur site ou compatible Amazon RDS for MySQL dans votre configuration de migration AWS DMS. Ces scripts fonctionnent avec un point de terminaison source ou cible. Les scripts sont tous écrits pour être exécutés sur la ligne de commande SQL MySQL.

Pour en savoir plus sur l'installation du client MySQL, consultez Installation du shell MySQL (langue française non garantie) dans la documentation MySQL. Pour en savoir plus sur l'utilisation du client MySQL, consultez Utilisation des commandes du shell MySQL (langue française non garantie) dans la documentation MySQL.

Avant d'exécuter un script, assurez-vous que le compte d'utilisateur que vous utilisez dispose des autorisations nécessaires pour accéder à la base de données compatible MySQL. Utilisez la procédure suivante pour créer un compte d'utilisateur et fournir les autorisations minimales requises pour exécuter ce script.

Pour configurer un compte d'utilisateur avec les autorisations minimales nécessaires pour exécuter ces scripts

1. Créez l'utilisateur pour exécuter les scripts.

```
create user 'username'@'hostname' identified by password;
```

2. Accordez la commande `select` sur les bases de données pour les analyser.

```
grant select on database-name.* to username;
grant replication client on *.* to username;
```

3.

```
grant execute on procedure mysql.rds_show_configuration to username;
```

Les rubriques suivantes décrivent comment télécharger, passer en revue et exécuter chaque script d'assistance disponible pour une base de données compatible MySQL. Elles décrivent également comment passer en revue et charger le résultat du script dans votre dossier AWS Support.

Rubriques

- Script awsdms_support_collector_MySQL.sql

Script awsdms_support_collector_MySQL.sql

Téléchargez le script `awsdms_support_collector_MySQL.sql`.

Ce script collecte des informations sur la configuration de la base de données compatible SQL Server. N'oubliez pas de vérifier la somme de contrôle sur le script et, si cette somme de contrôle est correcte, passez en revue le code SQL du script pour mettre en commentaires toute partie du code que vous n'envisagez pas sereinement d'exécuter. Une fois que vous êtes satisfait de l'intégrité et du contenu du script, vous pouvez exécuter ce dernier.

Exécutez le script après vous être connecté à votre environnement de base de données à l'aide de la ligne de commande.

Pour exécuter ce script et charger les résultats dans votre dossier d'assistance

1. Connectez-vous à la base de données à l'aide de la commande `mysql` suivante.

```
mysql -h hostname -P port -u username database-name
```

2. Exécutez le script à l'aide de la commande mysql `source` suivante.

```
mysql> source awsdms_support_collector_MySQL_compatible_DB.sql
```

Passez en revue le rapport généré et supprimez toutes les informations que vous ne souhaitez pas partager. Lorsque le contenu vous semble approprié à partager, chargez le fichier dans votre dossier AWS Support. Pour plus d'informations sur le chargement de ce fichier, consultez Utilisation de scripts d'assistance au diagnostic dans AWS DMS.

> **Note**
>
> - Si vous possédez déjà un compte d'utilisateur doté des privilèges requis décrits dans Scripts d'assistance au diagnostic pour les bases de données compatibles MySQL , vous pouvez également utiliser le compte d'utilisateur existant pour exécuter le script.
> - N'oubliez pas de vous connecter à la base de données avant d'exécuter le script.
> - Le script génère sa sortie au format texte.
> - En gardant à l'esprit les bonnes pratiques de sécurité, si vous créez un nouveau compte d'utilisateur uniquement pour exécuter ce script d'assistance au diagnostic MySQL, nous vous recommandons de supprimer ce compte d'utilisateur après l'exécution réussie du script.

Scripts d'assistance au diagnostic PostgreSQL

Vous trouverez ci-dessous les scripts d'assistance au diagnostic disponibles pour analyser un SGBDR PostgreSQL quelconque (sur site, Amazon RDS ou Aurora PostgreSQL) dans votre configuration de migration AWS DMS. Ces scripts fonctionnent avec un point de terminaison source ou cible. Les scripts sont tous écrits pour être exécutés dans l'utilitaire de ligne de commande psql.

Avant d'exécuter ces scripts, veillez à ce que le compte d'utilisateur que vous utilisez dispose des autorisations nécessaires suivantes pour accéder à tout SGBDR PostgreSQL :

- PostgreSQL version 10.x ou ultérieure : compte d'utilisateur avec autorisation d'exécution sur la fonction `pg_catalog.pg_ls_waldir`.

- PostgreSQL version 9.x ou antérieure : compte d'utilisateur avec autorisations par défaut.

Nous vous recommandons d'utiliser un compte existant disposant des autorisations appropriées pour exécuter ces scripts.

Si vous devez créer un nouveau compte d'utilisateur ou accorder des autorisations à un compte existant pour exécuter ces scripts, vous pouvez exécuter les commandes SQL suivantes pour tout SGBDR PostgreSQL basé sur la version de PostgreSQL.

Pour autoriser le compte à exécuter ces scripts pour une base de données PostgreSQL version 10.x ou ultérieure

- Effectuez l'une des actions suivantes :
  - Pour un nouveau compte d'utilisateur, exécutez ce qui suit.

```
CREATE USER script_user WITH PASSWORD 'password';
GRANT EXECUTE ON FUNCTION pg_catalog.pg_ls_waldir TO script_user;
```

  - Pour un compte d'utilisateur existant, exécutez ce qui suit.

```
GRANT EXECUTE ON FUNCTION pg_catalog.pg_ls_waldir TO script_user;
```

Pour autoriser le compte à exécuter ces scripts pour une base de données PostgreSQL version 9.x ou antérieure

- Effectuez l'une des actions suivantes :
  - Pour un nouveau compte d'utilisateur, exécutez ce qui suit avec les autorisations par défaut.

```
CREATE USER script_user WITH PASSWORD password;
```

  - Pour un compte d'utilisateur existant, utilisez les autorisations existantes.

> **Note**
>
> Ces scripts ne prennent pas en charge certaines fonctionnalités liées à la détermination de la taille de journal WAL pour les bases de données PostgreSQL versions 9.x et antérieures. Pour plus d'informations, utilisez AWS Support.

Les rubriques suivantes décrivent comment télécharger, passer en revue et exécuter chaque script d'assistance disponible pour PostgreSQL. Elles décrivent également comment passer en revue et charger la sortie des scripts dans votre dossier AWS Support.

Rubriques

- Script awsdms_support_collector_postgres.sql

Script awsdms_support_collector_postgres.sql

Téléchargez le script `awsdms_support_collector_postgres.sql`.

Ce script collecte des informations sur la configuration de la base de données PostgreSQL. N'oubliez pas de vérifier la somme de contrôle sur le script. Si la somme de contrôle est correcte, passez en revue le code SQL du script pour mettre en commentaires toute partie du code que vous n'envisagez pas sereinement d'exécuter. Une fois que vous êtes satisfait de l'intégrité et du contenu du script, vous pouvez exécuter ce dernier.

> **Note**
>
> Vous pouvez exécuter ce script avec le client psql version 10 ou ultérieure.

Vous pouvez utiliser les procédures suivantes pour exécuter ce script depuis votre environnement de base de données ou depuis la ligne de commande. Dans les deux cas, vous pourrez charger ultérieurement votre fichier sur AWS Support.

Pour exécuter ce script et charger les résultats dans votre dossier d'assistance

1. Effectuez l'une des actions suivantes :

   - Exécutez le script à partir de votre environnement de base de données à l'aide de la ligne de commande psql suivante.

```
dbname=# \i awsdms_support_collector_postgres.sql
```

À l'invite suivante, entrez le nom d'un seul des schémas que vous souhaitez migrer.

À l'invite suivante, entrez le nom de l'utilisateur (*script_user*) que vous avez défini pour vous connecter à la base de données.

- Exécutez le script suivant directement à partir de la ligne de commande. Cette option permet d'éviter toute invite préalable à l'exécution du script.

```
psql -h database-hostname -p port -U script_user -d database-name -f
 awsdms_support_collector_postgres.sql
```

2. Passez en revue le fichier HTML de sortie et supprimez toutes les informations que vous ne souhaitez pas partager. Lorsque le fichier HTML vous semble approprié à partager, chargez-le dans votre dossier AWS Support. Pour plus d'informations sur le chargement de ce fichier, consultez Utilisation de scripts d'assistance au diagnostic dans AWS DMS.

Utilisation de l'AMI d'assistance au diagnostic AWS DMS

Si vous rencontrez un problème lié au réseau lorsque vous utilisez AWS DMS, votre ingénieur support peut avoir besoin de plus d'informations sur la configuration de votre réseau. Nous voulons nous assurer qu'AWS Support obtienne autant d'informations que possible dans les plus brefs délais. Par conséquent, nous avons développé une AMI Amazon EC2 préconçue avec des outils de diagnostic pour tester votre environnement réseau AWS DMS.

Les tests de diagnostic installés sur l'Amazon Machine Image (AMI) incluent les suivants :

- Virtual Private Cloud (VPC)
- Perte de paquets réseau
- Latence du réseau
- Taille de l'unité de transmission maximale (MTU)

Rubriques

- Lancement d'une nouvelle instance Amazon EC2 de diagnostic AWS DMS
- Créer un rôle IAM

- Exécution de tests de diagnostic
- Étapes suivantes
- ID d'AMI par région

> **Note**
>
> Si vous rencontrez des problèmes de performances avec votre source Oracle, vous pouvez évaluer les performances de lecture de vos journaux redo ou d'archivage Oracle afin de trouver des moyens d'améliorer les performances. Pour plus d'informations, consultez Évaluation des performances de lecture des journaux redo ou d'archivage Oracle.

Lancement d'une nouvelle instance Amazon EC2 de diagnostic AWS DMS

Dans cette section, vous allez lancer une nouvelle instance Amazon EC2. Pour plus d'informations sur la façon de lancer une instance Amazon EC2, consultez Didacticiel : démarrez avec les instances Linux Amazon EC2 dans le Guide de l'utilisateur Amazon EC2.

Lancez une instance Amazon EC2 avec les paramètres suivants :

- Pour Images d'applications et de systèmes d'exploitation (Amazon Machine Image), recherchez l'AMI DMS-DIAG-AMI. Si vous êtes connecté à la console, vous pouvez rechercher cette AMI à l'aide de cette requête. Pour connaître l'ID d'AMI de l'AMI de diagnostic AWS dans votre région, consultez ID d'AMI par région ci-dessous.
- Pour Type d'instance, nous vous recommandons de choisir t2.micro.
- Pour Paramètres réseau, choisissez le même VPC que celui utilisé par votre instance de réplication.

Une fois l'instance active, connectez-vous y. Pour en savoir plus sur la connexion à une instance Amazon EC2 Linux, consultez Connectez-vous à votre instance Linux.

Créer un rôle IAM

Si vous souhaitez exécuter les tests de diagnostic sur votre instance de réplication en utilisant les autorisations minimales requises, créez un rôle IAM qui utilise la politique d'autorisations suivante :

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "dms:DescribeEndpoints",
                "dms:DescribeTableStatistics",
                "dms:DescribeReplicationInstances",
                "dms:DescribeReplicationTasks",
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "*"
        }
    ]
}
```

Attachez ce rôle à un nouvel utilisateur IAM. Pour en savoir plus sur la création de rôles, de politiques et d'utilisateurs IAM, consultez les sections suivantes du Guide de l'utilisateur IAM :

- Démarrer avec IAM
- Création de rôles IAM
- Création de politiques IAM

Exécution de tests de diagnostic

Après avoir créé une instance Amazon EC2 et vous y être connecté, procédez comme suit pour exécuter des tests de diagnostic sur votre instance de réplication.

1. Configurez l'interface de ligne de commande AWS :

   ```
   $ aws configure
   ```

   Fournissez les informations d'identification d'accès pour le compte d'utilisateur AWS que vous souhaitez utiliser pour exécuter les tests de diagnostic. Indiquez la région pour votre VPC et votre instance de réplication.

2. Affichez les tâches AWS DMS disponibles dans votre région. Remplacez l'exemple de région par votre région.

```
$ dms-report -r us-east-1 -l
```

Cette commande affiche le statut de vos tâches.

3. Affichez les points de terminaison et les paramètres des tâches. Remplacez *<DMS-Task-ARN>* par l'Amazon Resource Name (ARN) de votre tâche.

```
$ dms-report -t <DMS-Task-ARN>
```

Cette commande affiche les points de terminaison et les paramètres de votre tâche.

4. Exécutez des tests de diagnostic. Remplacez*<DMS-Task-ARN>* par l'ARN de votre tâche.

```
$ dms-report -t <DMS-Task-ARN> -n y
```

Cette commande affiche des données de diagnostic concernant le VPC de votre instance de réplication, la transmission de paquets réseau, la latence du réseau et la taille de l'unité de transmission maximale (MTU) du réseau.

```
########
#
#
#       AWS DMS Diagnostic
#       Date: 07-13-2022
#
#
#       aws region: us-east-2
#
#
########
==== DMS DIAG Info ====
Public IP: 3.
Private IP: 172.30.0.240
Instance ID: i-04829b2beb8214602
Instance MAC: 02:58:04:b5:52:28
Instane Type: t2.micro
Instance Sec Group: D
Instance AWS Region: us-east-2
Instance VPC Id: vpc-08ba020355d8a952e


==== Network Packet Check ====
1.) Check DMS EC2 MetaData service
>>>>>Result: 10 packets transmitted, 10 packets received, 0% packet loss
     Looks good with no issue. <<<<<

2.) Check Source endpoint (d
>>>>>Result: 10 packets transmitted, 10 packets received, 0% packet loss
     Looks good with no issue. <<<<<

3.) Check Target endpoint (r
>>>>>Result: 10 packets transmitted, 10 packets received, 0% packet loss
     Looks good with no issue. <<<<<

==== End network packet check ====


==== Network Latency Check ====
1.) Check DMS MetaData Service
>>>>>Result: round-trip min/avg/max = 0.4/0.4/0.5 ms
     Looks good with no issue. <<<<<

2.) Check Source endpoint (d
>>>>>Result: round-trip min/avg/max = 1.0/1.1/1.2 ms
     Looks good with no issue. <<<<<

3.) Check Target endpoint (r
>>>>>Result: round-trip min/avg/max = 1.4/1.4/1.5 ms
     Looks good with no issue. <<<<<

==== End network latency check ====


==== Network MTU Check ====
1.) Check DMS MetaData Service
>>>>>Result: MTU setting looks good. Local MTU (9001) matches remote MTU (9001) <<<<<


2.) Check Source endpoint (d
>>>>>Result: MTU setting looks good. Local MTU (9001) matches remote MTU (9001) <<<<<


3.) Check Target endpoint (r
>>>>>Result: MTU setting looks good. Local MTU (9001) matches remote MTU (9001) <<<<<


==== End network MTU check ====
```

Étapes suivantes

Les sections suivantes décrivent les informations de résolution de problèmes basées sur les résultats des tests de diagnostic réseau :

Test de VPC

Ce test vérifie que l'instance Amazon EC2 de diagnostic se trouve dans le même VPC que l'instance de réplication. Si l'instance Amazon EC2 de diagnostic ne figure pas dans le même VPC que votre instance de réplication, résiliez-la et créez-la à nouveau dans le VPC approprié. Vous ne pouvez pas modifier le VPC d'une instance Amazon EC2 après l'avoir créé.

Test de perte de paquets réseau

Ce test envoie 10 paquets aux points de terminaison suivants et vérifie si une perte de paquets s'est produite :

- Service de métadonnées Amazon EC2 d'AWS DMS sur le port 80
- Point de terminaison source
- Point de terminaison cible

Tous les paquets devraient arriver avec succès. En cas de perte de paquets, consultez un ingénieur réseaux pour déterminer le problème et trouver une solution.

Test de latence du réseau

Ce test envoie 10 paquets aux mêmes points de terminaison que le test précédent et vérifie la latence des paquets. Tous les paquets doivent avoir une latence inférieure à 100 millisecondes. Si un ou des paquets ont une latence supérieure à 100 millisecondes, consultez un ingénieur réseaux pour déterminer le problème et trouver une solution.

Test de taille de l'unité de transmission maximale (MTU)

Ce test détecte la taille de l'unité MTU en utilisant l'outil Traceroute sur les mêmes points de terminaison que ceux du test précédent. Tous les paquets du test doivent avoir la même taille d'unité MTU. Si des paquets ont une taille d'unité MTU différente, consultez un spécialiste du système pour déterminer le problème et trouver une solution.

ID d'AMI par région

Le tableau suivant indique l'ID d'AMI de l'AMI de diagnostic AMS dans votre région :

Région	ID de l'image
USA Est (Ohio)	ami-0f9c6801e8cab9399
USA Ouest (Oregon)	ami-01ceeab889d966e16
USA Est (Virginie du Nord)	ami-0617285c8257e8f6e
USA Ouest (Californie du Nord)	ami-0983b989d0a31ed55
Canada (Centre)	ami-0ba9380b30eee288e
Europe (Irlande)	ami-05959fe4fcc448f01
Europe (Londres)	ami-0de91dfa18b365c04
Europe (Paris)	ami-0844ecd26f2cc9073
Europe (Francfort)	ami-09bc69e6da9f3adf4
Europe (Stockholm)	ami-01da340b09bf5ab78
Europe (Milan)	ami-0dcfc51c17cfe7727
Europe (Espagne)	ami-0461c0b2e391c1280
Asie-Pacifique (Singapour)	ami-04150ffe00fe300d6
Asie-Pacifique (Sydney)	ami-049c1d1a9d72bafce
Asie-Pacifique (Jakarta)	ami-02189a4b8d3b2ee23
Asie-Pacifique (Hong Kong)	ami-0543ae06eff17cb07
Asie Pacifique (Tokyo)	ami-07b91e8e25d7b7890
Asie-Pacifique (Séoul)	ami-0657ad8cb9b5cf71c

Région	ID de l'image
Asie-Pacifique (Osaka)	ami-01ccf3e974933afdc
Asie-Pacifique (Mumbai)	ami-0d9fa24bbb8225e60
Moyen-Orient (EAU)	ami-0e3dcf51dfe58929b
Moyen-Orient (Bahreïn)	ami-0faad6f64101607f5
Amérique du Sud (São Paulo)	ami-097dd866777097d96

Référence AWS DMS

Cette section de référence comprend des informations supplémentaires dont vous pouvez avoir besoin lorsque vous utilisez AWS Database Migration Service (AWS DMS), y compris des informations de conversion de type de données.

AWS DMS conserve les types de données lorsque vous effectuez une migration de base de données homogène dans laquelle la source et la cible utilisent tous les deux le même type de moteur. Lorsque vous effectuez une migration hétérogène au cours de laquelle vous migrez des données d'un type de moteur de base de données vers un autre moteur de base de données, les types de données sont convertis en un type de données intermédiaire. Pour voir comment s'affichent les types de données sur la base de données cible, consultez les tables de types de données pour les moteurs de base de données source et cible.

Tenez compte de quelques points importants sur les types de données lors de la migration d'une base de données :

- Le type de données FLOAT est par essence une approximation. Lorsque vous insérez une valeur spécifique dans FLOAT, elle peut être représentée différemment dans la base de données. Cette différence est liée au fait que FLOAT n'est pas un type de données exact, comme un type de données décimales tel que NUMBER ou NUMBER(p,s). Par conséquent, la valeur interne des données FLOAT stockées dans la base de données peut être différente de la valeur que vous insérez. La valeur migrée d'une donnée FLOAT peut donc ne pas correspondre exactement à la valeur dans la base de données source.

  Pour plus d'informations sur ce problème, consultez les articles suivants :

  - Virgule flottante IEEE dans Wikipedia
  - Représentation de la virgule flottante IEEE sur Microsoft Learn
  - Pourquoi les nombres à virgule flottante peuvent manquer de précision sur Microsoft Learn

Rubriques

- Types de données pour AWS Database Migration Service

Types de données pour AWS Database Migration Service

AWS Database Migration Service utilise des types de données intégrés pour migrer les données d'un type de moteur de base de données source vers un type de moteur de base de données cible. Le tableau suivant illustre les types de données intégrées et leurs descriptions.

Types de données AWS DMS	Description
CHAÎNE	Chaîne de caractères.
WSTRING	Chaîne de caractères à deux octets.
BOOLEAN	Valeur booléenne.
BYTE	Valeur de données binaires.
DATE	Valeur de date : année, mois, jour.
TIME	Valeur temporelle : heure, minutes, secondes.
DATETIME	Valeur d'horodatage : année, mois, jour, heure, minute, seconde, fractions de secondes. Les fractions de secondes ont une échelle maximale de 9 chiffres. Le format suivant est pris en charge : AAAA:MM:JJ HH:MM:SS. F(9). Pour Amazon S3 Select et Amazon S3 Glacier Select, le format du type de données DATETIME est différent. Pour plus d'informations, consultez la description du type de données primitif `timestamp` dans Types de données pris en charge, dans le Guide de l'utilisateur Amazon Simple Storage Service.
INT1	Entier signé sur un octet.
INT2	Entier signé sur deux octets.
INT4	Entier signé sur quatre octets.

Types de données AWS DMS	Description
INT8	Entier signé sur huit octets.
NUMERIC	Valeur numérique exacte avec une précision et une échelle fixes.
REAL4	Valeur à virgule flottante simple précision.
REAL8	Valeur à virgule flottante double précision.
UINT1	Entier non signé sur un octet.
UINT2	Entier non signé sur deux octets.
UINT4	Entier non signé sur quatre octets.
UINT8	Entier non signé sur huit octets.
BLOB	Objet binaire volumineux.
CLOB	Grand objet de caractères.
NCLOB	Grand objet de caractères natif.

> **Note**
>
> AWS DMS ne peut pas migrer un type de données LOB vers un point de terminaison Apache Kafka.

AWS Notes de mise à jour du DMS

Vous trouverez ci-dessous les notes de publication pour les versions actuelles et précédentes de AWS Database Migration Service (AWS DMS).

AWS DMS ne fait pas la différence entre les versions majeures et secondaires lorsque vous activez la mise à niveau automatique des versions pour votre instance de réplication. DMS met automatiquement à niveau la version de l'instance de réplication pendant la période de maintenance si la version est obsolète.

Notez que pour mettre à niveau la version de votre instance de réplication manuellement (à l'aide de l'API ou de la CLI) de la version 3.4.x à la version 3.5.x, vous devez définir le paramètre sur. `AllowMajorVersionUpgrade` `true` Pour plus d'informations sur le `AllowMajorVersionUpgrade` paramètre, consultez ModifyReplicationInstancela documentation de l'API DMS.

> **Note**
>
> La version actuelle du moteur par défaut pour AWS DMS est 3.5.1.

AWS Notes de mise à jour du Database Migration Service 3.5.2

Nouvelles fonctionnalités de la version AWS DMS 3.5.2

Nouvelle fonction ou amélioration	Description
Validation des données Redshift	AWS DMS prend désormais en charge la validation des données dans les cibles Redshift.
Prise en charge de Microsoft SQL Server 2022 en tant que source et cible.	AWS DMS prend désormais en charge l'utilisation de Microsoft SQL Server version 2022 comme source et cible.
IBM Db2 LUW en tant que cible	AWS DMS prend désormais en charge IBM Db2 LUW en tant que cible. En utilisant AWS DMS, vous pouvez désormais effectuer des migrations en direct d'IBM Db2 LUW vers IBM Db2 LUW.

AWS Notes de mise à jour du Database Migration Service 3.5.1

Le tableau suivant présente les nouvelles fonctionnalités et améliorations introduites dans AWS Database Migration Service (AWS DMS) version 3.5.1.

Nouvelle fonction ou amélioration	Description
Prise en charge de PostgreSQL 15.x	AWS DMS la version 3.5.1 prend en charge la version 15.x de PostgreSQL. Pour plus d'informations, consultez Utilisation de PostgreSQL comme source et Utilisation de PostgreSQL comme cible.
Prise en charge des clusters élastiques Amazon DocumentDB avec des collections fragmentées	AWS DMS la version 3.5.1 prend en charge les clusters élastiques Amazon DocumentDB avec des collections fragmentées. Pour plus d'informations, consultez Utilisation d'Amazon DocumentDB en tant que cible pour AWS Database Migration Service.
Redshift sans serveur en tant que cible	Prise en charge de l'utilisation d'Amazon Redshift sans serveur en tant que point de terminaison cible. Pour plus d'informations, consultez Utilisation d'une base de données Amazon Redshift en tant que cible pour AWS Database Migration Service.
Paramètres de point de terminaison Babelfish	Paramètres de point de terminaison cible PostgreSQL améliorés pour assurer la prise en charge de Babelfish. Pour plus d'informations, consultez Utilisation d'une base de données PostgreSQL comme cible pour AWS Database Migration Service.
Transactions ouvertes de source Oracle	AWS DMS 3.5.1 améliore la méthodologie de gestion des transactions ouvertes lors du démarrage d'une tâche uniquement CDC à partir de la position de départ d'une source Oracle. Pour plus d'informations, consultez `OpenTransactionWindow` dans la section Paramètres du point de terminaison lors de l'utilisation d'Oracle comme source pour AWS DMS.
Amazon Timestream en tant que cible	Support pour l'utilisation d'Amazon Timestream comme point de terminaison cible. Pour plus d'informations, consultez Utilisation

Nouvelle fonction ou amélioration	Description
	d'Amazon Timestream comme cible pour AWS Database Migration Service.

AWS DMS la version 3.5.1 inclut les problèmes résolus suivants :

Problème résolu	Description
Oracle en tant que source de plus en plus de sessions inactives	Correction d'un problème pour la source Oracle où les tâches réservées au CDC entraînaient une augmentation continue du nombre de sessions inactives, ce qui entraînait l'exception suivante :. `ORA-00020: maximum number of processes exceeded on the source database`
Réplication des modifications de mise à jour dans DocumentDB	Correction d'un problème lié à DocumentDB en tant que cible, à cause duquel les instructions UPDATE n'étaient pas correctement répliquées dans certains scénarios.
Tâche de validation uniquement	Gestion des erreurs améliorée pour que la fonction de validation des données échoue correctement lorsque la validation des données est désactivée pour les tâches de validation uniquement.
Réplication Redshift après la fin de la connexion	Correction d'un problème pour la cible Redshift où la tâche DMS n'essayait pas à nouveau d'appliquer des modifications à la cible lorsque celle-ci avait `ParallelApplyThreads` défini une valeur supérieure à zéro après la fin de la connexion, ce qui entraînait une perte de données.
Réplication de texte en texte moyen MySQL	Correction d'un problème de réplication MySQL vers MySQL de types de données de type texte moyen en mode Lob complet.
La tâche CDC ne se réplique pas avec une rotation du secret	Correction d'un problème lié aux tâches DMS qui `BatchApplyEnabled` définissaient l'`true`endroit où DMS arrêterait de répliquer les données une fois le mot de passe modifié par Secrets Manager.

Problème résolu	Description
Problème de segmentation MongoDB/DocumentDB	Correction d'un problème pour la source MongoDB/DocDB où la segmentation de plage ne fonctionnait pas correctement lorsque la colonne de clé primaire contenait une valeur élevée.
Validation des données Oracle pour les valeurs numériques indépendantes	Correction d'un problème pour Oracle Target en raison duquel DMS reconnaissait une valeur de type de données non liées `NUMERIC` comme étant « a » `STRING` lors de la validation des données.
Validation des données SQL Server	Correction d'un problème pour les points de terminaison SQL Server où la validation des données DMS générait une instruction SQL non valide.
Segmentation automatique de MongoDB	Amélioration de la fonctionnalité de partitionnement automatique des données lors de la migration de documents en parallèle depuis MongoDB en tant que source.
Format Apache Parquet d'Amazon S3	Correction d'un problème permettant de visualiser avec Python et Apache Arrow C++ les fichiers Apache Parquet écrits dans S3 en tant que cible.
Gestion des DDL avec PostgreSQL en tant que source	Correction d'un problème lié à une source PostgreSQL : les opérations DDL non prises en charge n'étaient pas correctement ignorées.
Erreur de données PostgreSQL `timestamp tz`	Correction d'un problème lié aux migrations de PostgreSQL vers PostgreSQL : l'horodatage avec les données de fuseau horaire n'était pas correctement migré quand l'application par lots était activée pendant la CDC.
Échec de validation d'Oracle vers PostgreSQL	Correction d'un problème lié aux migrations d'Oracle vers PostgreSQL : la validation des données échouait pour le type de données NUMERIC(38,30).
Erreur de type de données étendu Oracle	Correction d'un problème lié à une source Oracle : le type de données varchar étendu était tronqué.

Problème résolu	Description
Combinaison d'opérateurs de filtre	Correction d'un problème lié à la fonctionnalité de filtrage des colonnes : l'opérateur de colonne Null ne pouvait pas être combiné avec d'autres types d'opérateurs.
Latence de CDC résultant d'une journalisation excessive.	Correction d'un problème lié à une source PostgreSQL : une journalisation excessive des avertissements du plug-in pglogical provoquait une latence de CDC source.
Gestion de la réplication bidirectionnelle de Create Table DDL	Correction d'un problème lié à la réplication bidirectionnelle de PostgreSQL vers PostgreSQL : la modification Create Table DDL n'était pas correctement répliquée.
Échec de CDC lors de l'utilisation de filtres	Correction d'un problème lié à la fonctionnalité de filtrage : la réplication CDC échouait.
Validation du nom d'hôte de l'autorité de certification pour les points de terminaison Kafka	Amélioration de la fonctionnalité des points de terminaison Kafka en ajoutant l'option permettant de désactiver la validation du nom d'hôte de l'autorité de certification (`SslEndpointIdentificationAlgorithm`).
Validation d'IBM Db2 LUW	Problème résolu : les types de données de date source, d'horodatage et d'heure de DB2 LUW n'étaient pas gérés correctement lors de la validation des données.
Validation S3	Correction d'un problème lié aux migrations de DB2 LUW vers S3 : la fonctionnalité de validation ne gérait pas correctement le type de données timestamp(0).
Échec du redémarrage de la tâche DMS	Correction d'un problème lié à la source PostgreSQL en raison duquel AWS DMS la tâche ne redémarrait pas et ne pouvait pas consommer d'événements relationnels lors de l'utilisation du plugin pglogical.
Validation du type de données HIERARCHY par SQL Server	Correction d'un problème lié à une source SQL Server : la validation du type de données HIERARCHY échouait.

Problème résolu	Description
Chaînes SQL Server avec caractères de contrôle	Correction d'un problème lié à une source SQL Server : les chaînes contenant des caractères de contrôle n'étaient pas répliquées correctement.
Redshift avec Secrets Manager	Correction d'un problème lié à une cible Redshift : les tests du point de terminaison échouaient lors de l'utilisation de Secrets Manager.
Incohérence ParallelL oadThreads des paramètre s MySQL	Correction d'un problème lié à une cible MySQL : le paramètre `ParallelLoadThreads` n'était pas correctement conservé après la modification des paramètres de tâche.
Erreur liée au mappage des types de données de PostgreSQL vers Oracle	Correction d'un problème lié aux migrations de PostgreSQL vers Oracle : la tâche échouait lors de la réplication du type de données TEXT vers le type de données VARCHAR2(2000).
Validation des données d'Oracle vers PostgreSQL	Correction d'un problème lié aux migrations d'Oracle vers PostgreSQL : la validation des données signalait des faux positifs lorsque des caractères NULL étaient répliqués sous forme de caractères ESPACE.
Source SQL Server dans la AlwaysOn configuration	Correction d'un problème lié à la AlwaysOn configuration de la source SQL Server, qui entraînait l'échec de la AWS DMS tâche lorsque le nom de la réplique ne correspondait pas exactement au nom du serveur réel.
Échec des tests de point de terminaison dans la source Oracle	Correction d'un problème lié à la source Oracle en raison duquel le test de connexion du point de AWS DMS terminaison échouait en raison de privilèges insuffisants lors de la récupération de l'ID de session Oracle (SID).
La CDC ne sélectionne pas de nouvelles tables	Correction d'un problème lié aux tâches de CDC uniquement : les tables créées sur la source après le démarrage de la tâche n'étaient pas répliquées dans certains cas.
Transactions ouvertes dans Oracle en tant que source	Amélioration la méthodologie de gestion des transactions ouvertes lors du démarrage d'une tâche de CDC uniquement à partir de la position de départ pour une source Oracle.

Problème résolu	Description
Problème de données manquantes	Problème résolu : données manquantes lors de la reprise d'une tâche si celle-ci était arrêtée après l'application des modifications mises en cache (option `StopTaskCachedChangesApplied` définie sur true). Ce problème peut se produire rarement si les modifications mises en cache sur le disque de l'instance de AWS DMS réplication AWS DMS persistent en raison d'un volume élevé de modifications sur la source.
Problème de validation des données lié au type de données étendu	Correction d'un problème lié à la validation des données de PostgreSQL vers Oracle : la validation échouait pour les types de données étendus.
Problème de validation des données lors d'un codage de caractères incohérent	Correction d'un problème lié à la validation des données de SQL Server vers PostgreSQL : la validation échouait lorsque le codage des caractères n'était pas cohérent entre la source et la cible.
Problème de validation des données ORA-01455	Problème résolu : une erreur ORA-01455 se produisait au cours de la validation lors du mappage d'un `integer` PostgreSQL sur un `number(10)` Oracle.
Prise en charge de la propriété `IDENTITY` SQL Server	Correction d'un problème lié à la réplication de données de SQL Server vers SQL Server : la migration des colonnes d'identité échouait quand la colonne cible avait la propriété `IDENTITY`.
Problème de jeu de caractères avec les instructions ALTER	Correction d'un problème de réplication MySQL vers MySQL qui AWS DMS changeait le jeu de caractères en UTF16 lors de la migration d'une `ALTER` instruction pendant le CDC.
Prise en charge du type de données spatial de PostgreSQL vers Redshift	Ajout de la prise en charge du type de données `spatial` lors de la migration de PostgreSQL vers Amazon Redshift.
Compression GZIP des fichiers .parquet	Correction d'un problème qui AWS DMS empêchait de générer des fichiers .parquet avec une compression GZIP avec S3 comme cible.

Problème résolu	Description
Migration des sources MongoDB/DocDB	Correction d'un problème qui AWS DMS empêchait la migration de certaines partitions depuis une source MongoDB.
Problème de statistiques de table	Problème résolu : les statistiques de table n'étaient pas affichées lorsqu'au moins une des tâches de l'instance de réplication contenait plus de 1001 tables.
Table suspendue pour IBM Db2 LUW versions 10.1.0 et antérieures	Correction d'un problème lié à une source DB2 LUW : la migration des tables était suspendue avec l'erreur `TYPESTRINGUNITS is not valid` quand la version de la base de données source était 10.1.0 ou antérieure.
Problème de partition nement MongoDB	Correction d'un problème lié à MongoDB/DocDB : un ou plusieurs segments de la partition source manquaient.
Problème de partition nement MongoDB	Correction d'un problème d'échec de la segmentation basée sur une colonne de type NumberLong () en raison d'un bogue de conversion de type.
Problème de partition nement MongoDB	Performances de segmentation automatique améliorées pour les jeux de données volumineux avec MongoDB en tant que source.
Version de pilote MongoDB	Le pilote MongoDB a été rétrogradé à la version 1.20.0 pour continuer à prendre en charge MongoDB versions 3.6 et antérieur es.
Type de données d'horodat age Amazon S3 Apache Parquet	Correction d'un problème pour Amazon S3 Parquet Target. AWS DMS définit désormais le paramètre de format `isAdjuste dToUTC` de manière `true` à ce qu'il corresponde au comportem ent des versions précédentes de AWS DMS.
Commande de copie d'Amazon Redshift en tant que cible	Correction d'un problème lié à Amazon Redshift en tant que cible : la commande de copie échouait pour des tables volumineuses lors de la copie de données d'Amazon S3 vers Amazon Redshift.

Problème résolu	Description
Types de données de géométrie PostgreSQL	Correction d'un problème lié aux migrations de PostgreSQL vers PostgreSQL : la migration échouait sur des types de données de géométrie volumineuse.
XML d'Oracle vers PostgreSQL	Problème résolu : la migration ajoutait un espace supplémentaire sur le XML lors de la réplication d'Oracle vers PostgreSQL.
Mise à jour du point de contrôle cible dans les moteurs pris en charge	AWS DMS met désormais à jour le point de contrôle cible dans la `awsdms_txn_state` table de la base de données cible.
Enregistrements MongoDB/DocDB envoyés à la mauvaise collection	Correction d'un problème lié à MongoDB/DocDB : les données étaient envoyées à la mauvaise collection cible.
Sélection d'une nouvelle table source Oracle avec paramètre de point de EscapeCharacter terminaison	Correction d'un problème lié à la source Oracle qui ne AWS DMS récupérait de nouvelles tables à répliquer que lorsque la tâche était arrêtée et reprise alors que le paramètre du `EscapeCharacter` point de terminaison était défini.
Point de contrôle de récupération CDC	Incohérence corrigée dans le point de contrôle de récupération CDC observée entre l'entrepôt de données cible et la console AWS DMS .
Tâches de validation CDC uniquement	Correction d'un problème lié aux tâches de validation CDC uniquement : la tâche n'échouait pas, même si toutes les tables de la tâche rencontraient des échecs.
Comportement de la validation en cas de problèmes de connexion source ou cible	Correction d'un problème de validation des données qui AWS DMS entraînait la suspension des tables sur la source ou la cible en cas d'interruption de la connexion.

Problème résolu	Description
Faux positifs de validation des données d'Oracle vers PostgreSQL	Correction d'un problème lié à la validation AWS DMS des données entre Oracle et PostgreSQL qui signalait des faux positifs. Ce problème était dû aux différences de représentation des caractères NULL sources sur la cible qui n'étaient pas prises en compte avec des types de données textuels autres que VARCHAR.
Troncation des données d'Oracle vers PostgreSQL	Correction d'un problème avec Oracle en tant que source et PostgreSQL en tant que cible : AWS DMS tronquait les données des colonnes NVARCHAR avec le paramètre Oracle `NLS_NCHAR _CHARACTERSET` défini sur `AL16UTF16` .
Erreur de validation des données	Correction d'un problème lié à la validation des données : une erreur `unable to create where filter clause` était générée lorsque le filtrage de la source et une règle de transform ation d'ajout de colonne étaient utilisés.
Gestion des erreurs de cible Redshift	Correction d'un problème lié à Redshift en tant que cible : la gestion des erreurs ne fonctionnait pas comme elle avait été configurée lorsque le paramètre de tâche `ParallelApplyThreads` de la tâche CDC était défini sur une valeur supérieure à zéro.
Échec de communica tion d'Oracle en tant que source	Correction d'un problème lié à Oracle en tant que source : la tâche restait dans l'état `RUNNING` mais n'était pas en mesure de migrer les données après un échec de communication.
Table CDC suspendue avec filtres de colonnes	Correction d'un problème lié aux tâches Chargement complet + CDC : une table était suspendue pendant la phase CDC lorsque des filtres de colonnes étaient appliqués.
Échec de validation des données de S3 en tant que cible pour des caractères spéciaux	Correction d'un problème lié à la validation des données cibles S3 : la tâche échouait si le nom de la table incluait un caractère spécial autre qu'un trait de soulignement.

Problème résolu	Description
Échec du chargement complet et CDC de la source MongoDB	Correction d'un problème lié à MongoDB en tant que source : une tâche Chargement complet + CDC échouait lors de la gestion des événements de cache lors de la migration d'une collection volumineuse.
Problème de mise à niveau avec `BatchApplyEnabled` défini sur true	Problème résolu : dans certains cas, une tâche dont le paramètre de `BatchApplyEnabled` tâche était défini sur true échouait après la migration de AWS DMS la version 3.4.6 vers la version 3.5.1.
AlwaysOn Source SQL Server avec classement distinguant majuscules et minuscules	Correction d'un problème lié à SQL Server AlwaysOn en tant que source en raison duquel une tâche échouait en cas de classement distinguant majuscules et minuscules.
Suspension de tâche source MySQL	Correction d'un problème lié à MySQL en tant que source : une tâche était suspendue au lieu d'échouer lorsque la source n'était pas correctement configurée.
Échec de tâche de chargement complet de source S3	Correction d'un problème lié à S3 en tant que source, à cause duquel une tâche échouait à la reprise après la mise à niveau de AWS DMS la version 3.4.6 ou 3.4.7 vers la version 3.5.1.
Source PostgreSQL avec paramètres CaptureDDL définis sur false	Correction d'un problème lié à PostgreSQL en tant que source : les DDL n'étaient pas correctement gérées lorsque le paramètre de point de terminaison `CaptureDDLs` était défini sur false.
Blocage de la tâche source Oracle pendant la reprise	Correction d'un problème lié à Oracle en tant que source : une tâche s'interrompait à sa reprise en raison de données incorrectes dans le nom de colonne.
Échec de la recherche d'objets LOB dans la source MySQL	Correction d'un problème lié à MySQL en tant que source : une recherche d'objets LOB échouait lorsque le paramètre de tâche `ParallelApplyThreads` était défini sur une valeur supérieure à zéro.

Problème résolu	Description
Erreur LSN illogique dans la source SQL Server	Correction d'un problème lié à SQL Server en tant que source, à cause duquel une tâche échouait avec une `illogical LSN sequencing state error` erreur après la mise à niveau de AWS DMS la version 3.4.7 vers la version 3.5.1.
Source PostgreSQL avec pglogical	Correction d'un problème lié à PostgreSQL en tant que source : une tâche utilisant le plug-in pglogical échouait quand la tâche était arrêtée, qu'une table était supprimée des règles de sélection, que la tâche était reprise et que des modifications étaient apportées à la table supprimée.
Point de contrôle de récupération Aurora MySQL incorrect.	Correction d'un problème lié à Aurora MySQL en tant que source : un point de contrôle de récupération incorrect était enregistré à la suite d'un basculement d'Aurora ou d'un arrêt et d'un redémarrage de la source Aurora.
SQL Server en tant que tâche source se bloque.	Correction d'un problème lié à SQL Server en tant que source : une tâche se bloquait lorsque `SafeguardPolicy` était défini sur `RELY_ON_SQL_SERVER_REPLICATION_AGENT` .
Conversion de type de données incorrecte avec MySQL comme cible	Correction d'un problème lié à MySQL en tant que cible : la réplicati on CDC échouait en raison d'une conversion de type de données incorrecte lors de la phase batch-apply.
Échec de tâche lorsque CaptureDDL était défini sur false pour PostgreSQL en tant que source.	Correction d'un problème lié à PostgreSQL en tant que source : une tâche échouait parce qu'un DDL était traité comme un DML lorsque le paramètre du point de terminaison `CaptureDDLs` était défini sur `false`.
Blocage pour cause de collection MongoDB vide	Correction d'un problème lié à MongoDB en tant que source : la tâche se bloquait en raison d'une collection vide.
Redshift en tant que cible : blocage d'une tâche à chargement complet	Correction d'un problème lié à Redshift en tant que cible : une tâche se bloquait pendant la phase de chargement complet lorsque la table de contrôle des points de contrôle de récupération était activée.

Problème résolu	Description
S3 vers S3 : pas de mouvement de données.	Correction d'un problème de réplication S3 vers S3 qui AWS DMS empêchait de répliquer les données si elles n'`bucketFolder` étaient pas spécifiées.
Latence de la CDC avec `GlueCatalogGeneration` défini sur `true`	Correction d'un problème lié à S3 en tant que cible : une latence excessive se produisait lorsque `GlueCatalogGeneration` était défini sur `true`.
Oracle en tant que cible : troncature des données	Correction d'un problème lié à Oracle en tant que cible où les données étaient AWS DMS tronquées dans les colonnes VARCHAR2.
Comportement du caractère générique « _ » dans PostgreSQL	Correction d''un problème lié à PostgreSQL en tant que source : le comportement du caractère générique « _ » dans les règles de sélection ne fonctionnait pas comme indiqué.
PostgreSQL en tant que source : problème d'en-tête WAL vide.	Correction d'un problème lié à PostgreSQL en tant que source : la tâche échouait en raison d'un en-tête WAL vide reçu de l'emplacement de réplication.
MySQL ou MariaDB en tant que source avec des journaux binaires compressés	Correction d'un problème pour MySQL et MariaDB en tant que sources où aucun message d'erreur approprié n'était émis AWS DMS lors de la détection d'une compression BINLOG.
Validation des données S3 : caractères spéciaux	Amélioration de la validation des données S3 pour gérer les caractères spéciaux dans les colonnes de clés primaires et non primaires.
Redshift en tant que cible : entrées trompeuses dans le journal des tâches	Correction d'un problème lié à Redshift en tant que cible : des entrées trompeuses présentes dans le journal des tâches signalaient l'échec de l'instruction batch-apply sur UPDATES et DELETES.
Blocage de la tâche de migration de SQL Server vers S3	Correction d'un problème lié aux migrations de SQL Server vers S3 : la tâche se bloquait lors de l'application des modifications mises en cache.

Problème résolu	Description
Données manquantes sur les erreurs batch-apply	Correction d'un problème lié à la fonctionnalité batch-apply : une erreur dans l'application d'un lot entraînait une absence de données.

AWS Notes de mise à jour de Database Migration Service 3.5.0 bêta

> **Important**
>
> AWS DMS 3.5.0 est une version bêta du moteur d'instance de réplication. AWS DMS supporte cette version de la même manière que toutes les versions précédentes. Mais nous vous recommandons de tester la version AWS DMS 3.5.0 bêta avant de l'utiliser à des fins de production.

Le tableau suivant présente les nouvelles fonctionnalités et améliorations introduites dans la version 3.5.0 bêta de AWS Database Migration Service (AWS DMS).

Nouvelle fonction ou amélioration	Description
Voyage dans le temps pour Oracle et Microsoft SQL Server	Vous pouvez désormais utiliser Time Travel dans toutes les AWS régions avec des points de terminaison sources Oracle, Microsoft SQL Server et PostgreSQL compatibles avec le DMS, et des points de terminaison cibles PostgreSQL et MySQL compatibles avec le DMS.
Validation S3	AWS DMS prend désormais en charge la validation des données répliquées dans les points de terminaison cibles Amazon S3. Pour en savoir plus sur la validation des données cibles Amazon S3, consultez Validation des données cibles Amazon S3.
Intégration du catalogue Glue	AWS Glue est un service qui fournit des méthodes simples pour classer les données et consiste en un référentiel de métadonnées

Nouvelle fonction ou amélioration	Description
	appelé AWS Glue Data Catalog. Vous pouvez désormais intégrer un AWS Glue Data Catalog à votre point de terminaison cible Amazon S3 et interroger les données Amazon S3 via d'autres AWS services tels qu'Amazon Athena. Pour plus d'informations, consultez Utilisation d'AWS Glue Data Catalog avec une cible Amazon S3 pour AWS DMS.
Application parallèle pour DocumentDB en tant que cible	L'utilisation de DocumentDB comme cible avec les nouveaux paramètres de `ParallelApply*` tâche permet AWS DMS désormais de prendre en charge un maximum de 5 000 enregistr ements par seconde lors de la réplication CDC. Pour plus d'informa tions, consultez Utilisation d'Amazon DocumentDB en tant que cible pour AWS Database Migration Service.
Journalisation centrée sur le client	Vous pouvez désormais examiner et gérer les journaux des tâches de manière plus efficace avec AWS DMS la version 3.5.0. Pour plus d'informations sur l'affichage et la gestion des journaux des tâches AWS DMS, consultezAffichage et gestion des journaux de tâches AWS DMS.
Mécanisme SASL_PLAI N pour les points de terminaison cibles Kafka	Vous pouvez désormais utiliser l'authentification SASL_PLAIN pour prendre en charge les points de terminaison cibles MSK Kafka.
Réplication des transacti ons XA dans MySQL	Vous pouvez désormais utiliser les transactions XA sur votre source DMS MySQL. Avant DMS 3.5.0, les modifications DML appliquée s dans le cadre des transactions XA n'étaient pas correctement répliquées.
Types de données étendus Oracle	AWS DMS prend désormais en charge la réplication de types de données étendus dans Oracle version 12.2 et supérieures.

Nouvelle fonction ou amélioration	Description
Environnement DB2 LUW PureScale	AWS DMS prend désormais en charge la réplication à partir d'un environnement DB2 LUW. PureScale Cette fonctionnalité est prise en charge uniquement à l'aide de l'option Commencer le traitement des modifications à partir de la position de modification de source.
Source SQL Server avec l'option `READ_COMMITTED_SNAPSHOT`	Lorsque vous utilisez une base de données source Microsoft SQL Server dont l'`READ_COMMITTED_SNAPSHOT` option est définie sur`TRUE`, vous pouvez répliquer correctement les modifications DML en définissant l'attribut de connexion forceDataRowLookup.

AWS DMS La version 3.5.0 inclut les problèmes résolus suivants :

Problèmes résolus dans la AWS DMS version 3.5.0 lancée le 17 mars 2023

Rubrique	Résolution
Oracle : comparaison d'un cas spécial pour une chaîne qui a été convertie à partir du format numérique	Correction d'un problème lié à une source Oracle : les règles de filtrage ne fonctionnaient pas comme prévu pour une colonne numérique lorsqu'une transform ation de type de données en chaîne existait pour cette colonne.
Améliorations du groupe de disponibi lité SQL Server sur site	Efficacité améliorée de la gestion des connexions avec la source SQL Server dans AlwaysOnla configuration en éliminant les connexions inutiles aux répliques qui ne sont pas utilisées par DMS.
Conversion interne HIERARCHYID de SQL Server	Correction d'un problème lié à une source SQL Server : le type de données HIERARCHYID était répliqué en tant que VARCHAR(250) au lieu de HIERARCHYID vers la cible SQL Server.
Correction de la tâche de déplacement de la cible S3	Problème résolu : le déplacement d'une tâche avec une cible S3 prenait beaucoup de temps, semblait figé ou ne se terminait jamais.

Rubrique	Résolution	
Mécanisme SASL Plain de Kafka	Introduction de la prise en charge de la méthode d'authentification SASL Plain pour le point de terminaison cible MSK Kafka.	
Le chargement/l'application en parallèle échoue à cause du paramètre _type avec Opensearch 2.x	Correction d'un problème lié à une cible Opensearch 2.x : le chargement parallèle ou l'application parallèle échouaient en raison de l'absence de prise en charge du paramètre _type.	
Prise en charge du filtre de mappage de tables avec opérateurs mixtes	Suppression d'une limitation selon laquelle un seul filtre pouvait être appliqué à une colonne.	
Points de terminaison S3, Kinesis et Kafka : migration des colonnes lob basée sur Alter dans la phase CDC	Correction d'un problème lié aux cibles Kinesis, Kafka et S3 : les données des colonnes LOB ajoutées pendant la CDC n'étaient pas répliquées.	
Mise à niveau du pilote MongoDB	Le pilote MongoDB a été mis à niveau vers v1.23.2.	
Mise à jour du pilote Kafka	Le pilote Kafka a été mis à jour de la version 1.5.3 à la version 1.9.2.	
Fonctionnement incorrect d'un paramètre de point de terminaison S3	Correction d'un problème lié à une cible S3 : le paramètre de point de terminaison `AddTrailingPaddingCharacter` ne fonctionnait pas lorsque les données contenaient le caractère spécifié comme séparateur pour la cible S3.	

Rubrique	Résolution	
Blocage de la tâche cible Kinesis	Correction d'un problème lié à une cible Kinesis : une tâche s'interrompait quand la valeur PK était vide et que le débogage détaillé était activé.	
Décalage d'une position des noms des colonnes des cibles S3	Correction d'un problème lié à une cible S3 : les noms des colonnes étaient décalés d'une position quand `AddColumnName` était défini sur `true` et `TimestampColumnName` était défini sur `""`.	
Journalisation améliorée des avertissements de troncation de LOB	Amélioration de la journalisation des avertissements en cas de troncation du LOB pour la source SQL Server afin d'inclure l'instruction select utilisée pour récupérer le LOB.	
Ajout d'une erreur fatale pour éviter les interruptions des tâches DMS si le mot de passe TDE est erroné	Introduction d'un message d'erreur significatif et élimination du problème de défaillance de tâche dans les situations où la tâche DMS échouait sans message d'erreur en raison d'un mot de passe TDE incorrect pour Oracle en tant que source.	
Autorisation de la migration de la DDL PostgreSQL CTAS (Create Table As Select) pendant la CDC.	Des limitations ont été supprimées qui empêchaient DMS de répliquer les DDL PostgreSQL CTAS (Create Table As Select) pendant la CDC	
Correction de la défaillance de la tâche pg_logical lors de la suppression de colonnes d'une table pendant la CDC	Correction d'un problème lié à une source PostgreSQL avec cible S3 : les colonnes étaient mal alignées sur la cible lorsque la prise en charge des LOB était désactivé e et que des LOB étaient présents.	

Rubrique	Résolution	
Correction d'une fuite de mémoire dans la gestion des connexions MySQL	Correction d'un problème lié à une source MySQL : la consommation de mémoire des tâches augmentait continuellement.	
Paramètre de point de terminaison dans la source Oracle : `ConvertTimestampWithZoneToUTC`	Définissez cet attribut sur `true` pour convertir la valeur d'horodatage des colonnes « TIMESTAMP WITH TIME ZONE » et « TIMESTAMP WITH LOCAL TIME ZONE » au format UTC. Par défaut, la valeur de cet attribut est « false » et les données sont répliquées en utilisant le fuseau horaire de la base de données source.	
Source Oracle : `DataTruncationErrorPolicy` défini sur SUSPEND_TABLE ne fonctionne pas	Correction d'un problème lié à une source Oracle avec cible S3 : les tables n'étaient pas suspendues alors que le paramètre de tâche `DataTruncationErrorPolicy` était défini sur SUSPEND_TABLE.	
Échec SQL Server sur un schéma long/une table longue lors de la génération de la clause de requête	Correction d'un problème lié à une source SQL Server : une tâche échouait ou ne répondait plus lorsque la règle de sélection contenait une liste de tables séparées par des virgules.	
Authentification Secret Manager avec le point de terminaison MongoDB	Correction d'un problème lié aux points de terminaison MongoDB et DocumentDB : l'authentification basée sur le gestionnaire de secrets ne fonctionnait pas.	

Rubrique	Résolution	
Troncation par DMS des données pendant la CDC pour une colonne varchar multi-octets quand NLS_NCHAR _CHARACTERSET est défini sur UTF8	Correction d'un problème lié à une source Oracle avec cible Oracle : les données étaient tronquées pour les colonnes VARCHAR multi-octets avec NLS_NCHAR _CHARACTERSET défini sur UTF8.	
`filterTra nsactions 0fUser` ECA pour Oracle LogMiner	Ajout d'un attribut de connexion supplémentaire (ECA) `filterTransactions0fUser` pour permettre à DMS d'ignorer les transactions d'un utilisateur spécifié lors de la réplication depuis Oracle à l'aide de. LogMiner	
Erreur récupérab le de paramètre SQL Server quand un numéro LSN est absent de la sauvegarde	Correction d'un problème lié à SQL Server : une tâche n'échouait pas en cas d'absence de numéro LSN.	

AWS Notes de mise à jour du Database Migration Service 3.4.7

Le tableau suivant présente les nouvelles fonctionnalités et améliorations introduites dans la version 3.4.7 AWS de Database Migration Service (AWS DMS).

Nouvelle fonction ou amélioration	Description
Prise en charge de Babelfish en tant que cible	AWS DMS prend désormais Babelfish en tant que cible. En utilisant AWS DMS, vous pouvez désormais migrer des données en direct depuis n'importe quelle source AWS DMS prise en charge vers un Babelfish, avec un minimum de temps d'arrêt.

Nouvelle fonction ou amélioration	Description
	Pour plus d'informations, consultez Utilisation de Babelfish en tant que cible pour AWS Database Migration Service.
Prise en charge des bases de données IBM Db2 z/OS en tant que source pour le chargement complet uniquement	AWS DMS prend désormais en charge les bases de données IBM Db2 z/OS en tant que source. En utilisant AWS DMS, vous pouvez désormais effectuer des migrations en direct depuis les mainframes DB2 vers n'importe quelle cible AWS DMS prise en charge. Pour plus d'informations, consultez Utilisation de bases de données IBM Db2 for z/OS en tant que source pour AWS DMS.
Prise en charge des réplicas de lecture SQL Server en tant que source	AWS DMS prend désormais en charge la réplication en lecture de SQL Server en tant que source. En utilisant AWS DMS, vous pouvez désormais effectuer des migrations en direct depuis SQL Server Read Replica vers n'importe quelle cible AWS DMS prise en charge. Pour plus d'informations, consultez Utilisation d'une base de données Microsoft SQL Server comme source pour AWS DMS.
Support aux EventBridge événements DMS	AWS DMS prend en charge la gestion des abonnements aux événements à EventBridge l'aide d'événements DMS. Pour plus d'informations, consultez Utilisation des événements et des notifications Amazon EventBridge dans AWS Database Migration Service.

Nouvelle fonction ou amélioration	Description
Prise en charge des points de terminaison sources et cibles de VPC	AWS DMS prend désormais en charge les points de terminaison Amazon Virtual Private Cloud (VPC) en tant que sources et cibles. AWS DMS peuvent désormais se connecter à n'importe quel AWS service doté de points de terminaison VPC lorsque des routes explicitement définies vers les services sont définies dans leur VPC. AWS DMS **Note** Les mises à niveau vers AWS DMS les versions 3.4.7 et supérieures nécessitent que vous configuriez d'abord AWS DMS pour utiliser des points de terminaison VPC ou pour utiliser des routes publiques. Cette exigence s'applique aux points de terminaison source et cible pour Amazon S3, Amazon Kinesis Data Streams AWS Secrets Manager, Amazon DynamoDB, Amazon Redshift et Amazon Service. OpenSearch Pour plus d'informations, consultez Configuration de points de terminaison de VPC en tant que points de terminaison sources et cibles AWS DMS.
Nouvelle version de PostgreSQL	PostgreSQL version 14.x est désormais pris en charge en tant que source et en tant que cible.
Prise en charge d'Aurora sans serveur v2 en tant que cible	AWS DMS prend désormais en charge Aurora Serverless v2 en tant que cible. En utilisant AWS DMS, vous pouvez désormais effectuer des migrations en direct vers Aurora Serverless v2. Pour plus d'informations sur AWS DMS les cibles prises en charge, consultezCibles pour la migration des données.

Nouvelle fonction ou amélioration	Description
Nouvelles versions d'IBM Db2 pour LUW	AWS DMS prend désormais en charge les versions 11.5.6 et 11.5.7 d'IBM Db2 for LUW en tant que source. En utilisant AWS DMS, vous pouvez désormais effectuer des migrations en direct à partir des dernières versions d'IBM DB2 for LUW. Pour plus d'informations sur AWS DMS les sources, voirSources pour la migration des données. Pour plus d'informations sur AWS DMS les cibles prises en charge, consultezCibles pour la migration des données.

AWS DMS La version 3.4.7 inclut les comportements nouveaux ou modifiés suivants et les problèmes résolus suivants :

- Vous pouvez désormais utiliser un format de date issu de la définition de table pour analyser une chaîne de données en un objet de date lorsque vous utilisez Amazon S3 en tant que source.
- De nouveaux compteurs de statistiques de tables sont désormais disponibles : `AppliedInserts`, `AppliedDdls`, `AppliedDeletes` et `AppliedUpdates`.
- Vous pouvez désormais choisir le type de mappage par défaut lorsque vous l'utilisez OpenSearch comme cible.
- Le nouveau paramètre de point de terminaison `TrimSpaceInChar` pour les sources Oracle, PostgreSQL et SQLServer vous permet de spécifier s'il convient de découper les données sur les types de données CHAR et NCHAR.
- Le nouveau paramètre de point de terminaison `ExpectedBucket0wner` pour Amazon S3 empêche toute tricherie lors de l'utilisation de S3 en tant que source ou cible.
- Pour RDS SQL Server, Azure SQL Server et SQL Server autogéré, DMS fournit désormais une configuration automatique de MS-CDC sur toutes les tables sélectionnées pour une tâche de migration avec ou sans CLÉ PRIMAIRE, ou avec un index unique tenant compte de la priorité d'activation de la réplication Microsoft sur les tables SQL Server autogérées avec CLÉ PRIMAIRE.
- Ajout de la prise en charge de la réplication des opérations DDL de partition et de sous-partition Oracle lors des migrations Oracle homogènes.

- Problème résolu : une tâche de validation de données s'interrompait avec une clé primaire composite lors de l'utilisation d'Oracle en tant que source et cible.
- Correction d'un problème lié à la conversion correcte d'un type de caractères variable en booléen alors que la colonne cible était précédemment créée en tant que booléen lors de l'utilisation de Redshift en tant que cible.
- PCorrection d'un problème qui provoquait la troncation des données pour les types de données `varchar` migrés en `varchar(255)` en raison d'un problème ODBC connu lors de l'utilisation de PostgreSQL en tant que cible.
- Problème résolu : l'indicateur parallèle pour l'opération DELETE n'était pas respecté avec `BatchApplyEnabled` défini sur `true` et `BatchApplyPreserveTransaction` défini sur `false` lors de l'utilisation d'Oracle en tant que cible.
- Le nouveau paramètre de point de terminaison `AddTrailingPaddingCharacter` d'Amazon S3 ajoute un remplissage aux données de chaîne lors de l'utilisation de S3 en tant que cible.
- Le nouveau paramètre de tâche `max_statement_timeout_seconds` prolonge le délai d'expiration par défaut des requêtes de point de terminaison. Ce paramètre est actuellement utilisé par les requêtes de métadonnées de point de terminaison MySQL.
- Lors de l'utilisation de PostgreSQL en tant que cible, un problème a été résolu : une tâche de CDC n'utilisait pas correctement les paramètres de la tâche de gestion des erreurs.
- Problème résolu : DMS ne pouvait pas identifier correctement le mode Redis pour une instance Redis Enterprise.
- La prise en charge de l'attribut de connexion supplémentaire (ECA) `includeOpForFullLoad` a été étendue pour le format Parquet cible S3.
- Introduction d'un nouveau paramètre de point de terminaison PostgreSQL `migrateBooleanAsBoolean`. Lorsque ce paramètre est défini sur `true` pour une migration de PostgreSQL vers Redshift, un booléen sera migré comme varchar(1). Lorsqu'il est défini sur `false`, un booléen est migré comme varchar(15), ce qui est le comportement par défaut.
- Lors de l'utilisation d'une source SQL Server, un problème de migration lié au type de données `datetime` a été résolu. Cette correction résout le problème de l'insertion de `Null` lorsque la précision est exprimée en millisecondes.
- Pour une source PostgreSQL avec PGLOGICAL, un problème de migration a été résolu lors de l'utilisation de pglogical et de la suppression d'un champ de la table source pendant la phase CDC : la valeur après le champ supprimé n'était pas migrée vers la table cible.
- Correction d'un problème lié à la migration de SQL Server Loopback : la réplication bidirectionnelle obtenait des enregistrements répétés.

- Ajout d'un nouvel attribut ECA `mapBooleanAsBoolean` pour PostgreSQL en tant que source. À l'aide de cet attribut de connexion supplémentaire, vous pouvez remplacer le mappage de type de données par défaut d'un booléen PostgressQL vers un type de données booléen. RedShift
- Correction d'un problème lié à la migration lors de l'utilisation de SQL Server en tant que source : ALTER DECIMAL/NUMERIC SCALE ne se répliquait pas sur les cibles.
- Correction d'un problème de connexion avec SQL Server 2005.
- Depuis le 17 octobre 2022, DMS 3.4.7 prend désormais en charge les classes d'instances Amazon EC2 de génération 6 pour les instances de réplication.
- Depuis le 25 novembre 2022, avec DMS 3.4.7, vous pouvez convertir des schémas de base de données et des objets de code en utilisant la conversion de schéma DMS, et découvrir les bases de données de votre environnement réseau qui sont de bons candidats à la migration en utilisant DMS Fleet Advisor.
- À compter du 25 novembre 2022, DMS Studio est retiré.
- Depuis le 31 janvier 2023, la conversion de schéma DMS prend en charge Aurora MySQL et Aurora PostgreSQL en tant que fournisseur de données cible.
- Depuis le 6 mars 2023, vous pouvez générer des recommandations cibles de taille adaptée pour vos bases de données sources avec DMS Fleet Advisor.
- Depuis le 6 mars 2023, AWS DMS prend en charge la politique AWS gérée qui permet de publier des points de données métriques sur Amazon CloudWatch.

Problèmes résolus dans la publication de maintenance de DMS 3.4.7 datée du 5 mai 2023

Rubrique	Résolution	
Échec des tâches pour une source PostgreSQL	Correction d'un problème lié à une source PostgreSQL : les tâches échouaient en cas de dépassement du nombre maximal d'opérations DDL autorisées dans un même événement.	
Faux positifs de validation des données d'une source PostgreSQL	Correction d'un problème lié à une source PostgreSQL avec cible Oracle : une conversio n incorrecte du champ d'horodatage entraînait des erreurs de validation des données de faux positifs.	

Rubrique	Résolution
Gestion des erreurs d'une source MySQL	Correction d'un problème lié à une source MySQL : la tâche DMS n'échouait pas lorsque le prochain journal BIN n'était pas disponible.
Journalisation de ROTATE_EVENT d'une source MySQL	Journalisation améliorée pour une source MySQL liée à ROTATE_EVENT : le nom du journal BIN en cours de lecture a été inclus.
Problème de délai de validation des données	Correction d'un problème lié à la fonctionnalité de validation des données : le paramètre de point de terminaison `executeTimeout` n'était pas respecté pour les requêtes liées à la validation des données.
Problème de chargement complet parallèle d'une cible PostgreSQL	Correction d'un problème lié à une cible PostgreSQL : le chargement complet segmenté (parallèle) échouait en raison d'une erreur de type « connexion interrompue ».
Problème de déplacement de tâches DMS	Correction d'un problème lié à une cible S3 : une opération de déplacement de tâches DMS prenait beaucoup de temps ou ne se terminait jamais.
Problème d'enregistrement dupliqué d'une source PostgreSQL	Correction d'un problème lié à une source PostgreSQL : une tâche DMS générait des erreurs liées à des doublons sur la cible après l'arrêt et la reprise d'une tâche.
Faux positifs de validation des données d'une cible Oracle	Correction d'un problème lié à une cible Oracle : la validation des données signalait des erreurs de faux positifs dues à une réplication incorrecte du fuseau horaire pour les champs d'horodatage.

Problèmes résolus dans la publication de maintenance de DMS 3.4.7 datée du 22 février 2023

Rubrique	Résolution	
Réplicas de groupe de disponibilité SQL Server en tant que source	Ajout de la prise en charge de la source SQL Server dans les AlwaysOnconfigurations où le port TCP de l'écouteur était différent du port TCP de réplique.	
Perte de données avec Amazon Redshift en tant que cible	Correction d'un problème lié à une cible Redshift : dans de rares cas, un redémarrage inattendu de Redshift pouvait entraîner des données manquantes sur la cible.	
Prise en charge de la protection de la source SQL Server	Correction d'un problème lié à une source SQL Server : la tâche DMS pouvait échouer avec une erreur indiquant l'impossibilité de lire les sauvegardes du journal des transactions lorsque le paramètre de point de terminaison `"SafeguardPolicy": "EXCLUSIVE_AUTOMATIC_TRUNCATION"` était spécifié.	
Échec de la tâche de validation des données pour Oracle en tant que source	Correction d'un problème lié à une source Oracle : la tâche DMS pouvait échouer lors de la validation des données en raison de valeurs de clé primaire mal identifiées.	
Problème de données d'image avec Kinesis	Correction d'un problème lié aux cibles de streaming (Kinesis, Kafka) : le paramètre de tâche `"EnableBeforeImage"` fonctionnait uniquement pour les types de données de caractère.	
Fichiers journaux de voyage dans le temps	Correction d'un problème lié à la fonctionnalité Voyage dans le temps : DMS créait des fichiers journaux de voyage dans le temps de zéro octet lorsque la source était inactive.	

Problèmes résolus dans la publication de maintenance de DMS 3.4.7 datée du 16 décembre 2022

Rubrique	Résolution	
BatchApplyEnabled	Correction d'un problème de journalisation excessive lorsque le BatchApplyEnabled paramètre est défini sur True.	
Nouveau paramètre de point de terminaison MongoDB : FullLoadNoCursorTimeout	Le paramètre du point de terminaison MongoDB FullLoadNoCursorTimeout indique NoCursorTimeout le curseur de chargement complet. NoCursorTimeout est un paramètre de connexion MongoDB qui empêche le serveur de fermer le curseur en cas d'inactivité.	
MongoDB : fonction de filtre pour la segmentation sur une seule colonne	La nouvelle fonction de filtre améliore les performances de migration des bases de données MongoDB en utilisant une seule colonne pour la segmentation.	
De MongoDB à Redshift	Lors de la migration de MongoDB vers Redshift, si la collection MongoDB contient des données de type binaire, un problème a été résolu : DMS ne créait pas la table cible sur Redshift.	
Nouvel attribut de connexion MongoDB SocketTimeout MS	Le nouvel attribut de connexion supplémentaire MongoDB SocketTimeout MS configure le délai de connexion pour les clients MongoDB en millisecondes. Si la valeur est inférieure ou égale à zéro, la valeur par défaut du client MongoDB est utilisée.	
Correction d’un problème à l’origine du blocage d’une tâche Amazon Kinesis	Lors de la migration vers Amazon Kinesis Data Streams en tant que cible, un problème a été résolu avec la gestion des valeurs null en l’absence de clé primaire dans la table.	
Prise en charge de la validation des données Oracle NULL PK/UK	Une limitation a été supprimée en raison de laquelle la validation des données des valeurs NULL PK/UK n’était pas prise en charge.	

Rubrique	Résolution	
D'Oracle à Amazon S3	Lors de la migration d'Oracle vers Amazon S3, un problème a été résolu : certains enregistrements étaient migrés de façon incorrecte en tant que valeurs NULL.	
Oracle Standby	Lors de l'utilisation d'Oracle Standby en tant que source, la possibilité pour DMS de gérer les transacti ons ouvertes a été ajoutée.	
Migration d'Oracle vers Oracle avec le type de données spatiales SDO_GEOMETRY	Lors de la migration d'Oracle vers Oracle, un problème a été résolu : la tâche échouait si la table avait une colonne SDO_GEOMERY présente dans le DDL.	
Oracle en tant que source	Lors de l'utilisation d'Oracle en tant que source, un problème a été résolu : DMS ignorait parfois un numéro de séquence de journal redo Oracle.	
Oracle en tant que source : journaux redo en ligne/d'archivage manquants	Lors de l'utilisation d'Oracle en tant que source, un problème a été résolu qui empêchait la tâche DMS d'échouer si des journaux d'archivage étaient manquants.	
Problème corrigé : DMS ignorait parfois un journal redo Oracle Standby	Lors de l'utilisation d'Oracle en tant que source, un problème a été résolu : DMS ignorait parfois un numéro de séquence de journal redo Oracle.	
Problème corrigé : non-réplication des types de données spatiales d'Oracle vers Oracle pendant la CDC	Lors de la réplication d'Oracle vers Oracle, un problème a été résolu : les types de données spatiales n'étaient pas répliqués pendant la CDC.	

Rubrique	Résolution	
Oracle en tant que cible	Lors de l'utilisation d'Oracle en tant que cible, un problème a été résolu : l'application cible échouait avec l'erreur ORA-01747.	
Amazon S3 : perte de données corrigée lors du rechargement de table	Lors de l'utilisation d'Amazon S3 en tant que cible, un problème a été résolu : une opération de rechargement de table ne générait pas de fichiers CDC.	
Problème corrigé : initialisation du contexte SQL Server AlwaysOn en cas d'utilisation du serveur principal en tant que source	Lors de l'utilisation de SQL Server Always On comme source, résolution d'un problème qui empêchait d'initialiser les groupes de disponibilité (AG) si la source est principale et si elle AlwaysOnSharedSyncedBackupIsEnabled est définie sur true.	
Mise à jour du paramètre de point de terminaison SQL Server	Lorsqu'un point de terminaison source est le groupe de disponibilité SQL Server Always On et qu'il s'agit d'une réplique secondaire, le problème selon lequel la tâche de réplication échouait AlwaysOnSharedSyncedBackupsIsEnabled s'il était défini sur True a été résolu.	
PostgreSQL en tant que source	Correction d'un problème en raison duquel le CDC ne parvenait pas à migrer les opérations de suppression/mise à jour sur la source PostgreSQL, introduit dans la version 3.4.7 pour prendre en charge le booléen. mapBooleanAs	

AWS Notes de mise à jour du Database Migration Service 3.4.6

Le tableau suivant présente les nouvelles fonctionnalités et améliorations introduites dans la version 3.4.6 AWS de Database Migration Service (AWS DMS).

Nouvelle fonction ou amélioration	Description
AWS DMS Voyage dans le temps	AWS DMS introduit Time Travel, une fonctionnalité qui offre aux clients une flexibilité dans leurs capacités de journalisation et améliore leur expérience de dépannage. Avec Time Travel, vous pouvez stocker et chiffrer les AWS DMS journaux à l'aide d'Amazon S3, et les consulter, les télécharger et les masquer dans un certain laps de temps.
Prise en charge de Microsoft Azure SQL Managed Instance en tant que source	AWS DMS prend désormais en charge l'instance Microsoft Azure SQL Managed Instance en tant que source. En utilisant AWS DMS, vous pouvez désormais effectuer des migrations en direct depuis une instance gérée Microsoft Azure SQL vers n'importe quelle cible AWS DMS prise en charge. Pour plus d'informations sur AWS DMS les sources, voirSources pour la migration des données. Pour plus d'informations sur AWS DMS les cibles prises en charge, consultezCibles pour la migration des données.
Prise en charge de Google Cloud SQL pour MySQL en tant que source	AWS DMS prend désormais en charge Google Cloud SQL pour MySQL en tant que source. En utilisant AWS DMS, vous pouvez désormais effectuer des migrations en direct depuis Google Cloud SQL pour MySQL vers n'importe quelle cible AWS DMS prise en charge. Pour plus d'informations sur AWS DMS les sources, voirSources pour la migration des données. Pour plus d'informations sur AWS DMS les cibles prises en charge, consultezCibles pour la migration des données.
Prise en charge du chargement parallèle pour les données partitionnées vers S3	AWS DMS prend désormais en charge le chargement parallèle des données partitionnées vers Amazon S3, ce qui améliore les temps de chargement pour la migration des données partitionnées depuis les données sources des moteurs de base de données

Nouvelle fonction ou amélioration	Description
	compatibles vers Amazon S3. Cette fonctionnalité crée des sous-dossiers Amazon S3 pour chaque partition de la table dans la source de base de données, ce qui permet à AWS DMS d'exécuter des processus parallèles pour remplir chaque sous-dossier.
Prise en charge de plusieurs rubriques cibles Apache Kafka en une seule tâche	AWS DMS supporte désormais les cibles multithématiques d'Apache Kafka avec une seule tâche. Avec AWS DMS, vous pouvez désormais répliquer plusieurs schémas d'une même base de données vers différentes rubriques cibles Apache Kafka en utilisant la même tâche. Cela élimine la nécessité de créer plusieurs tâches distinctes dans le cas où de nombreuses tables de la même base de données source doivent être migrées vers différentes rubriques cibles Kafka.

Les problèmes résolus dans la version AWS DMS 3.4.6 sont les suivants :

- Problème résolu : les colonnes des instructions UPDATE étaient remplies avec des colonnes incorrectes si la colonne de clé primaire n'était pas la première colonne lors de l'utilisation d'Amazon S3 en tant que cible au format CSV.
- Correction d'un problème à cause duquel AWS DMS des tâches pouvaient se bloquer lors de l'utilisation du plugin pglogical avec des `NULL` valeurs dans des `BYTEA` colonnes en mode LOB limité lors de l'utilisation de PostgreSQL comme source.
- Correction d'un problème en raison duquel AWS DMS les tâches pouvaient se bloquer lorsqu'un grand nombre de tables sources étaient supprimées lors de l'utilisation de PostgreSQL comme source.
- Le partitionnement de dossier basé sur la date Amazon S3 a été amélioré en introduisant un nouveau paramètre Amazon S3 `DatePartitionTimezone` permettant le partitionnement à des dates non-UTC.
- Prise en charge du mappage entre le type de données `TIMESTAMP WITH TIME ZONE` des sources et `TIMESTAMPTZ` lors de l'utilisation de Redshift en tant que cible
- Les performances de la CDC ont été améliorées pour les tâches sans règles de sélection de caractères génériques lors de l'utilisation de MongoDB ou d'Amazon DocumentDB en tant que source.

- Problème résolu : les noms de schéma avec un caractère générique de trait de soulignement et une longueur inférieure à 8 n'étaient pas capturés par les tâches AWS DMS lors de l'utilisation de Db2 LUW en tant que source.
- Correction d'un problème à cause duquel les AWS DMS instances manquaient de mémoire lorsque le volume de données était important lorsqu'elles utilisaient OpenSearch Service comme cible.
- Les performances de validation des données ont été améliorées en prenant en charge les tâches de validation de chargement complet uniquement.
- Correction d'un problème en raison duquel AWS DMS les tâches ne reprenaient pas après un basculement forcé lors de l'utilisation de Sybase comme source.
- Correction d'un problème en raison duquel un avertissement était AWS DMS envoyé `Invalid BC timestamp was encountered in column` incorrectement.

Les problèmes résolus dans la publication de maintenance de DMS 3.4.6 incluent les suivants :

- Problème résolu : une tâche s'interrompait quand le mode d'application en bloc était activé lors de l'utilisation d'Oracle en tant que source et cible.
- Problème résolu qui empêchait une tâche de chargement complet d'utiliser correctement le paramètre de point de terminaison `ExecuteTimeout` avec PostgreSQL en tant que source.
- Correction d'un problème lié à la migration des colonnes de type de données Array lorsque la tâche est définie sur le mode LOB limité tout en utilisant PostgreSQL en tant que source.
- Correction d'un problème lié à la migration des horodatages avec le fuseau horaire avant le 01/01/1970 lors de l'utilisation de PostgreSQL en tant que source.
- Problème résolu : DMS traitait une chaîne vide comme une valeur Null au cours de la réplication lors de l'utilisation de SQL Server en tant que source et cible.
- Correction d'un problème lié au respect des paramètres de point de terminaison de délai de lecture et d'écriture de session lors de l'utilisation d'une source/cible MySQL.
- Problème résolu : une tâche DMS CDC téléchargeait des fichiers associés au chargement complet lors de l'utilisation d'Amazon S3 en tant que source.
- Correction d'un problème de blocage de journalisation lorsque `CdcInsertsAndUpdates` et `PreserveTransactions` étaient définis sur `true` lors de l'utilisation d'Amazon S3 en tant que cible.
- Correction d'un problème en raison duquel une tâche se bloquait lorsque la ParallelApply fonctionnalité* était activée, mais que certaines tables ne disposaient pas de clé primaire par défaut lors de l'utilisation d'Amazon Kinesis Data Streams comme source.

- Correction d'un problème en raison duquel aucune erreur n'était indiquée pour une erreur StreamArn lors de l'utilisation d'Amazon Kinesis Data Streams comme source.
- Correction d'un problème à cause duquel une valeur de clé primaire sous forme de chaîne vide provoquait le blocage d'une tâche lorsqu'elle OpenSearch était utilisée comme cible.
- Problème résolu : trop d'espace disque était utilisé pour la validation des données.

Problèmes résolus dans la publication de maintenance de DMS 3.4.6 datée du 13 décembre 2022

Rubrique	Résolution
Pilote ODBC SAP ASE	Correction d'un problème lié à SAP ASE en tant que source afin que le pilote ODBC puisse prendre en charge les jeux de caractères.
Bogue de clé primaire de type datetime de Sqlserver pour la recherche d'objets LOB	Correction d'un problème lié à SQL Server en tant que source : la recherche d'objets LOB ne fonctionnait pas correctement pour une clé primaire de type datetime, avec une précision en millisecondes.
De SQL Server vers Redshift : « datetimeoffset » mappé sur « timestamptz »	Pour les migrations de SQL Server vers Redshift, le mappage a été amélioré afin que le format « datetimeoffset » de SQL Server soit mappé vers le format « timestamptz » de Redshift.
Validation des données : SkipLobColumns c'est vrai	Correction d'un problème en raison duquel la tâche DMS se bloque lorsque la valeur SkipLobColumns est True, qu'un LOB figure sur la source, que la clé primaire se trouve dans la dernière colonne et qu'une différence de données est détectée par validation.
Validation des données avec MySQL en tant que source	Correction d'un problème lié à MySQL en tant que source avec la validation des données activée : un blocage de tâche DMS se produisait lors de l'utilisation d'une table contenant une clé unique composite contenant des valeurs null.

Rubrique	Résolution	
MySQL en tant que source	Correction d’un problème lié à MySQL en tant que source : une table était suspendue avec une erreur de dépassement de capacité lorsque les colonnes étaient modifiées pour augmenter la précision.	
Mise à niveau du pilote ODBC MySQL vers la version 8.0.23	Correction d’un problème lié à MySQL en tant que source : le classement « utf8mb4_0900_bin » était incompatible avec le pilote MySQL utilisé par DMS.	
MySQL : prise en charge des modifications de DDL pour les tables partitionnées	Un nouveau paramètre de point de terminaison MySQL a été introduit pour permettre skipTableSuspension ForPartitionDdl à l'utilisateur d'ignorer la suspension des tables en cas de modification du DDL de partition pendant le CDC, de sorte que DMS peut désormais prendre en charge les modifications DDL pour les tables MySQL partitionnées.	
Migration de MongoDB vers Redshift	Correction d’un problème lié aux migrations de MongoDB vers Redshift : DMS ne parvenait pas à créer la table cible sur Redshift si la collection MongoDB contenait des données de type binaire.	
Cible Redshift : erreur de segmentation de voyage dans le temps pour l’application en bloc	Correction d'un problème lié à Redshift en tant que cible, à cause duquel la tâche DMS se bloquait alors qu'elle était définie sur true. BatchApplyEnabled	
Redshift en tant que cible	Correction d’un problème lié à Redshift en tant que cible : lorsque le chargement parallèle était défini sur type=partitions-auto, les segments parallèles écrivaient des fichiers CSV en bloc dans le même répertoire de table et interféraient les uns avec les autres.	

Rubrique	Résolution
Redshift en tant que cible	Correction d'un problème lié à Redshift en tant que cible : pendant la CDC, la colonne cible était de type booléen alors que la source était de type character varying.
Redshift en tant que cible	Le journal des tâches a été amélioré pour identifier une modification de DDL qui ne peut pas être répliquée vers Redshift en tant que cible.
Validation de données avec PostgreSQL	Correction d'un problème lié à la validation avec PostgreSQL : la validation échouait lorsque des types de données booléens étaient présents.
PostgreSQL en tant que source	Correction d'un problème lié à PostgreSQL en tant que source, de sorte que le chargement complet utilisait le champ dans Attributs ExecuteTimeout de connexion supplémentaires.
PostgreSQL en tant que source	Correction d'un problème lié à PostgreSQL en tant que source, afin qu'une tâche échoue si elle lit des LSN supérieurs au LSN de reprise de la tâche demandée de plus de 60 minutes pour indiquer l'existence d'un problème lié à l'emplacement de réplication utilisé.
PostgreSQL en tant que source : timestamptz avant le 01/01/1970	Correction d'un problème lié à PostgreSQL en tant que source : les données timestamptz avant le 01/01/1970 n'étaient pas migrées correctement pendant la CDC.
PostgreSQL en tant que source	Correction d'un problème lié à PostgreSQL en tant que source : DMS tronquait des valeurs de type character varying pendant la CDC.

Rubrique	Résolution	
PostgreSQL en tant que source : reprise de la tâche arrêtée	Correction d’un problème lié à PostgreSQL en tant que source : une ou plusieurs transactions étaient manquantes pendant la CDC lors de la reprise d’une tâche précédemment arrêtée.	
Amazon S3 en tant que cible	Correction d'un problème lié à S3 en tant que cible, à cause duquel l'en-tête du fichier CSV obtenu était décalé d'une colonne lorsqu' AddColumnName il était vrai et qu' TimestampColumnName il était « ».	
Amazon S3 en tant que source : comportement d’utilisation de la mémoire pendant la phase de chargement complet de la tâche	Correction d’un problème lié à S3 en tant que source : une tâche DMS dans le cadre du chargement complet ne libérait la mémoire utilisée qu’après le chargement de la table entière dans la base de données cible.	
Amazon S3 en tant que cible : opération de rechargement de table	Correction d’un problème lié à S3 en tant que cible : une opération de rechargement de table ne générait pas les fichiers CDC.	

AWS Notes de mise à jour du Database Migration Service 3.4.5

Le tableau suivant présente les nouvelles fonctionnalités et améliorations introduites dans la version 3.4.5 AWS de Database Migration Service (AWS DMS).

Nouvelle fonction ou amélioration	Description
Prise en charge de Redis en tant que cible	AWS DMS prend désormais en charge Redis en tant que cible. En utilisant AWS DMS, vous pouvez désormais migrer des données en direct depuis n'importe quelle source AWS DMS prise en charge vers un magasin de données Redis, avec un temps d'arrêt minimal.

Nouvelle fonction ou amélioration	Description
	Pour plus d'informations sur AWS DMS les cibles, consultezCibles pour la migration des données.
Prise en charge de MongoDB 4.2 et 4.4 en tant que sources	AWS DMS supporte désormais MongoDB 4.2 et 4.4 en tant que sources. Vous pouvez désormais migrer les données des clusters MongoDB 4.2 et 4.4 vers n'importe quelle cible AWS DMS prise en charge, y compris Amazon DocumentDB (compatible avec MongoDB), avec un temps d'arrêt minimal. AWS DMS Pour plus d'informations sur AWS DMS les sources, voirSources pour la migration des données.
Prise en charge de plusieurs bases de données utilisant MongoDB en tant que source	AWS DMS prend désormais en charge la migration de plusieurs bases de données en une seule tâche en utilisant MongoDB comme source. En utilisant AWS DMS, vous pouvez désormais regrouper plusieurs bases de données d'un cluster MongoDB et les migrer à l'aide d'une seule tâche de migration de base de données. Vous pouvez migrer vers n'importe quelle cible AWS DMS prise en charge, y compris Amazon DocumentDB (compatible avec MongoDB), avec un temps d'arrêt minimal.
Prise en charge de la segmentation automatique avec MongoDB ou Amazon DocumentDB (compatib le avec MongoDB) en tant que source	AWS DMS prend désormais en charge la segmentation automatiq ue en utilisant MongoDB ou Amazon DocumentDB comme source. À l'aide de AWS DMS, vous pouvez configurer des tâches de migration de base de données pour segmenter automatiquement la collection d'un cluster MongoDB ou DocumentDB. Vous pouvez ensuite migrer les segments en parallèle vers n'importe quelle cible AWS DMS prise en charge, y compris Amazon DocumentDB, avec un temps d'arrêt minimal.

Nouvelle fonction ou amélioration	Description
Amélioration des performances de chargement complet Amazon Redshift	AWS DMS prend désormais en charge l'utilisation de threads parallèles lors de l'utilisation d'Amazon Redshift comme cible pendant le chargement complet. En tirant parti des paramètres des tâches à chargement complet multithread, vous pouvez améliorer les performances de votre migration initiale depuis n'importe quelle source AWS DMS prise en charge vers Amazon Redshift. Pour plus d'informations sur AWS DMS les cibles, consultezCibles pour la migration des données.

Les problèmes résolus dans la version AWS DMS 3.4.5 sont les suivants :

- Problème résolu : les données pouvaient être manquantes ou dupliquées après reprise, lors de l'utilisation de PostgreSQL en tant que source avec un taux élevé de simultanéité des transactions.
- Problème résolu : les tâches de migration de base de données échouaient avec l'erreur Could not find relation id … lors de l'utilisation de PostgreSQL en tant que source, avec le plug-in pglogical activé.
- Problème résolu : les colonnes `VARCHAR` n'étaient pas répliquées correctement lors de l'utilisation de PostgreSQL en tant que source et d'Oracle en tant que cible.
- Problème résolu : les opérations de suppression n'étaient pas correctement capturées quand la clé primaire n'était pas la première colonne de la définition de table, lors de l'utilisation de PostgreSQL en tant que source.
- Problème résolu : les tâches de migration de base de données omettaient les mises à jour des objets LOB dans un paramètre de métadonnées spécial lors de l'utilisation de MySQL en tant que source.
- Problème résolu : les colonnes `TIMESTAMP` étaient traitées comme `DATETIME` en mode LOB complet lors de l'utilisation de MySQL version 8 en tant que source.
- Problème résolu : les tâches de migration de base de données échouaient lors de l'analyse d'enregistrements `NULL DATETIME` lors de l'utilisation de MySQL 5.6.4 ou version ultérieure en tant que source.
- Problème résolu : les tâches de migration de base de données se bloquaient après avoir rencontré une erreur Thread is exiting lors de l'utilisation d'Amazon Redshift en tant que cible avec une application parallèle.

- Problème résolu : une perte de données était possible quand les tâches de migration de base de données se déconnectaient d'un point de terminaison cible Amazon Redshift lors de l'application par lots de la CDC.
- Amélioration des performances de chargement complet en appelant `ACCEPTINVCHARS` lors de l'utilisation d'Amazon Redshift en tant que cible.
- Correction d'un problème en raison duquel les enregistrements dupliqués étaient répliqués lors du passage du mode one-by-one au mode d'application parallèle en utilisant Amazon Redshift comme cible.
- Problème résolu : les tâches de migration de base de données ne transféraient pas la propriété des objets Amazon S3 au propriétaire de compartiment avec `cannedAclForObjects=bucket_owner_full_control` lors de l'utilisation d'Amazon S3 en tant que cible.
- Amélioré AWS DMS grâce à la prise en charge de plusieurs destinations d'archivage avec ECA `additionalArchivedLogDestId` lors de l'utilisation d'Oracle comme source.
- Problème résolu : les tâches de migration de base de données échouaient avec l'erreur `OCI_INVALID_HANDLE` lors de la mise à jour d'une colonne LOB en mode LOB complet.
- Problème résolu : les colonnes `NVARCHAR2` n'étaient pas migrées correctement pendant la CDC lors de l'utilisation d'Oracle en tant que source.
- Amélioré AWS DMS par l'activation `SafeguardPolicy` lors de l'utilisation de RDS pour SQL Server en tant que source.
- Problème résolu : les tâches de migration de base de données signalaient une erreur sur `rdsadmin` lors de l'utilisation d'une source SQL Server autre que RDS.
- Problème résolu : la validation des données échouait avec l'UUID comme clé primaire dans un paramètre de partition lors de l'utilisation de SQL Server en tant que source.
- Problème résolu : les tâches de chargement complet + CDC pouvaient échouer si le LSN requis était introuvable dans le journal de base de données lors de l'utilisation de Db2 LUW en tant que source.
- Amélioré AWS DMS grâce à la prise en charge des horodatages CDC personnalisés lors de l'utilisation de MongoDB comme source.
- Problème résolu : les tâches de migration de base de données se bloquaient en s'arrêtant lors de l'utilisation de MongoDB en tant que source et en cas d'erreurs du pilote MongoDB sur `endSessions`.
- Correction d'un problème qui empêchait AWS DMS de mettre à jour les champs non principaux lors de l'utilisation de DynamoDB comme cible

- Problème résolu : la validation des données signalait des incohérences de faux positifs sur les colonnes `CLOB` et `NCLOB`.
- Problème résolu : la validation des données échouait sur les enregistrements contenant uniquement des espaces lors de l'utilisation d'Oracle en tant que source.
- Problème résolu : les tâches de migration de base de données s'interrompaient lors de la troncation d'une table partitionnée.
- Problème résolu : les tâches de migration de base de données échouaient lors de la création de la table de contrôle `awsdms_apply_exceptions`.
- Prise en charge étendue du plug-in d'authentification `caching_sha2_password` lors de l'utilisation de MySQL version 8.

AWS Notes de mise à jour du Database Migration Service 3.4.4

Le tableau suivant présente les nouvelles fonctionnalités et améliorations introduites dans AWS DMS version 3.4.4.

Nouvelle fonction ou amélioration	Description
Prise en charge du chiffrement TLS et de l'authentification TLS ou SASL en utilisant Kafka en tant que cible	AWS DMS prend désormais en charge le chiffrement TLS et l'authentification TLS ou SASL en utilisant Amazon MSK et le cluster Kafka sur site comme cible. Pour plus d'informations sur l'utilisation du chiffrement et de l'authentification pour les points de terminaison Kafka, consultez Connexion à Kafka à l'aide du protocole TLS (Transport Layer Security).

Les problèmes résolus dans la version AWS DMS 3.4.4 sont les suivants :

- Amélioration de la AWS DMS journalisation des échecs de tâches lors de l'utilisation des points de terminaison Oracle.
- L'amélioration de l'exécution des AWS DMS tâches poursuit le traitement lorsque les points de terminaison source Oracle changent de rôle après le basculement d'Oracle Data Guard.
- La gestion améliorée des erreurs traite ORA-12561 comme une erreur récupérable lors de l'utilisation de points de terminaison Oracle.

- Problème résolu : les colonnes `EMPTY_BLOB()` et `EMPTY_CLOB()` étaient migrées en tant que valeurs null lors de l'utilisation d'Oracle en tant que source.
- Correction d'un problème en raison duquel AWS DMS les tâches ne mettaient pas à jour les enregistrements après l'ajout de modifications DDL dans les colonnes lors de l'utilisation de SQL Server comme source.
- La migration de PostgreSQL en tant que source a été améliorée grâce à la prise en charge du type de données `TIMESTAMP WITH TIME ZONE`.
- Problème résolu : le paramètre `afterConnectScript` ne fonctionnait pas pendant un chargement complet lors de l'utilisation de PostgreSQL en tant que cible.
- Un nouveau paramètre `mapUnboundedNumericAsString` a été introduit pour mieux gérer le type de données `NUMERIC` sans précision ni échelle lors de l'utilisation des points de terminaison PostgreSQL.
- Correction d'un problème en raison duquel AWS DMS les tâches échouaient avec « aucune ligne affectée » après l'arrêt et la reprise de la tâche lors de l'utilisation de PostgreSQL comme source.
- Correction d'un problème qui AWS DMS empêchait la migration du type de `TIMESTAMP` données avec le `BC` suffixe lors de l'utilisation de PostgreSQL comme source.
- Correction d'un problème qui AWS DMS empêchait la migration de la `TIMESTAMP` valeur « ±infinity » lors de l'utilisation de PostgreSQL comme source.
- Problème résolu : les chaînes vides étaient traitées comme `NULL` si vous utilisiez S3 en tant que source avec le paramètre `csvNullValue` défini sur d'autres valeurs.
- L'attribut de connexion supplémentaire `timestampColumnName` a été amélioré dans le cadre d'un chargement complet avec CDC, afin de pouvoir être trié pendant la CDC lors de l'utilisation de S3 en tant que cible.
- Amélioration de la gestion des types de données binaires au format hexadécimal tels que `BYTE`, `BINARY` et `BLOB` lors de l'utilisation de S3 en tant que source.
- Problème résolu : les enregistrements supprimés étaient migrés avec des caractères spéciaux lors de l'utilisation de S3 en tant que cible.
- Problème résolu de gestion des valeurs de clé vides lors de l'utilisation d'Amazon DocumentDB (compatible avec MongoDB) en tant que cible.
- Correction d'un problème qui AWS DMS empêchait la réplication des `Decimal128` colonnes lors de l'utilisation de MongoDB `NumberDecimal` ou d'Amazon DocumentDB (avec compatibilité avec MongoDB) comme source.

- Problème résolu : il est désormais possible d'autoriser les tâches CDC à réessayer en cas de basculement sur MongoDB ou Amazon DocumentDB (compatible avec MongoDB) en tant que source.
- Ajout d'une option permettant de supprimer le préfixe hexadécimal « 0x » des valeurs de type de `RAW` données lors de l'utilisation de Kinesis, Kafka ou comme cible. OpenSearch
- Problème résolu : la validation échouait sur les colonnes de caractères de longueur fixe lors de l'utilisation de Db2 LUW en tant que source.
- Problème résolu : la validation échouait lorsque seul le type de données source ou le type de données cible était `FLOAT` ou `DOUBLE`.
- Problème résolu : la validation échouait sur les caractères `NULL` lors de l'utilisation d'Oracle en tant que source.
- Problème résolu : la validation échouait sur les colonnes XML lors de l'utilisation d'Oracle en tant que source.
- Correction d'un problème à cause duquel AWS DMS les tâches se bloquaient lorsqu'il y avait des colonnes nullables dans les clés composites utilisant MySQL comme source.
- Correction d'un problème qui empêchait de AWS DMS valider à la fois les `UNIQUEIDENTIFIER` colonnes des points de terminaison source de SQL Server et les colonnes UUID des points de terminaison cibles de PostgreSQL.
- Problème résolu : une tâche CDC n'utilisait pas une définition de table source mise à jour après sa modification.
- Amélioration du AWS DMS basculement pour traiter les échecs de tâches provoqués par un nom d'utilisateur ou un mot de passe non valide comme des erreurs récupérables.
- Correction d'un problème en raison duquel AWS DMS les tâches échouaient en raison de l'absence de LSN lors de l'utilisation de RDS pour SQL Server comme source.

AWS Notes de mise à jour du Database Migration Service 3.4.3

Le tableau suivant présente les nouvelles fonctionnalités et améliorations introduites dans AWS DMS version 3.4.3.

Nouvelle fonction ou amélioration	Description
Nouvelle version d'Amazon DocumentDB	Amazon DocumentDB version 4.0 est désormais pris en charge en tant que source.
Nouvelle version de MariaDB	MariaDB version 10.4 est désormais pris en charge en tant que source et cible.
Support à l' AWS Secrets Manager intégration	Vous pouvez stocker en toute sécurité les informations de connexion de base de données (informations d'identification utilisateur) pour les points de terminaison pris en charge dans AWS Secrets Manager. Vous pouvez ensuite envoyer le secret correspon dant au lieu des informations d'identification en texte brut AWS DMS lorsque vous créez ou modifiez un point de terminaison. AWS DMS se connecte ensuite aux bases de données des terminaux à l'aide du secret. Pour plus d'informations sur la création de secrets pour les AWS DMS points de terminaison, consultezUtilisation de secrets pour accéder aux points de terminaison AWS Database Migration Service.
Options étendues pour les instances de réplication C5 et R5	Vous pouvez désormais créer les instances de réplication de plus grande taille suivantes : C5 jusqu'à 96 vCPU et 192 Gio de mémoire et R5 jusqu'à 96 vCPU et 768 Gio de mémoire.
Amélioration des performances Amazon Redshift	AWS DMS prend désormais en charge l'application parallèle lorsque vous utilisez Redshift comme cible afin d'améliorer les performances de la réplication en cours. Pour plus d'informations, consultez Paramètres de tâche multithread pour Amazon Redshift.

Les problèmes résolus dans la AWS DMS version 3.4.3 sont les suivants :

- Problème résolu : l'horodatage de validation devenait « 1970-01-01 00:00:00 » pour les événements différés lors de l'utilisation de Db2 LUW en tant que source.
- Correction d'un problème en raison duquel AWS DMS les tâches échouaient avec une `NVARCHAR` colonne comme clé primaire lors de l'utilisation de SQL Server comme source en mode LOB complet.

- Problème résolu : enregistrements manquants au cours de la phase des modifications mises en cache lors de l'utilisation de SQL Server en tant que source.
- Correction d'un problème en raison duquel des enregistrements étaient ignorés après la reprise AWS DMS des tâches lors de l'utilisation de RDS pour SQL Server comme source.
- Correction d'un problème en raison duquel AWS DMS le composant de journalisation ASSERTION génère des journaux volumineux pour SQL Server.
- Problème résolu : la validation des données échouait pendant la phase CDC en raison d'un dépassement de l'analyse des colonnes lors de l'utilisation de MySQL en tant que source.
- Correction d'un problème en raison duquel AWS DMS les tâches se bloquaient en raison d'une erreur de segmentation lors de la validation des données lors de l'utilisation de PostgreSQL comme cible.
- Problème résolu : la validation des données échouait sur le type de données DOUBLE pendant la CDC lors de l'utilisation de PostgreSQL en tant que source et cible.
- Problème résolu : les enregistrements insérés par la commande copy n'étaient pas répliqués correctement lors de l'utilisation de PostgreSQL en tant que source et de Redshift en tant que cible.
- Problème de perte de données résolu pendant la phase des modifications mises en cache lors de l'utilisation de PostgreSQL en tant que source.
- Correction d'un problème qui pouvait provoquer une perte de données ou des doublons d'enregistrements lors de l'utilisation de PostgreSQL en tant que source.
- Problème résolu : les schémas présentant une casse mixte ne pouvaient pas migrer avec pglogical lors de l'utilisation de PostgreSQL en tant que source.
- Problème résolu : le dernier message d'échec ne contenait pas l'erreur ORA lors de l'utilisation d'Oracle en tant que source.
- Correction d'un problème en raison duquel AWS DMS les tâches ne parvenaient pas à générer des instructions UPDATE lors de l'utilisation d'Oracle comme cible.
- Correction d'un problème en raison duquel AWS DMS les tâches ne répliquaient pas les données lors de l'utilisation d'Oracle 12.2 comme source avec une configuration ASM et une base de données enfichable.
- Amélioration de l'analyse des enregistrements en préservant la conformité des guillemets à la RFC 4180 lors de l'utilisation de S3 en tant que source.
- Amélioration du traitement de `timestampColumnName` afin que la colonne provenant du chargement complet puisse être triée avec celle provenant de la CDC.

- En introduisant un nouveau paramètre de point de terminaison`MessageMaxBytes`, nous avons résolu un problème en AWS DMS raison duquel les tâches échouaient lorsque certains éléments LOB étaient supérieurs à 1 Mo.
- Correction d'un problème en raison duquel AWS DMS les tâches se bloquaient en raison d'une erreur de segmentation lors de l'utilisation de Redshift comme cible.
- Amélioration de la journalisation des erreurs pour la connexion de test Redshift.
- Correction d'un problème qui AWS DMS empêchait le transfert de tous les documents de MongoDB vers DocumentDB pendant le chargement complet.
- Correction d'un problème en raison duquel AWS DMS les tâches signalaient une erreur fatale lorsqu'aucune table n'était incluse dans les règles de mappage des tables.
- Problème résolu : les schémas et les tables créés avant le redémarrage des tâches AWS DMS n'étaient pas répliqués vers la cible lors de l'utilisation de MySQL en tant que source.
- Problème résolu : le caractère générique d'échappement [_] ne permettait pas d'échapper le caractère générique « _ » dans une règle d'exclusion lors de l'utilisation de MySQL en tant que source.
- Problème résolu : la colonne de type de données UNSIGNED BIGINT n'était pas répliquée correctement lors de l'utilisation de MySQL en tant que source.

AWS Notes de mise à jour du Database Migration Service 3.4.2

Le tableau suivant présente les nouvelles fonctionnalités et améliorations introduites dans AWS DMS version 3.4.2.

Nouvelle fonction ou amélioration	Description
Support pour la connexion privée de votre Amazon Virtual Private Cloud (Amazon VPC) au AWS Database Migration Service (DMS) sans passerelle Internet, appareil NAT, connexion	Vous pouvez désormais vous connecter à votre Amazon VPC et y accéder AWS DMS via un point de terminaison d'interface VPC que vous créez. Ce point de terminaison d'interface vous permet d'isoler toute l'activité réseau de votre instance de AWS DMS réplication au sein de l'infrastructure réseau Amazon. En incluant une référence à ce point de terminaison d'interface dans tous les appels d'API relatifs à l' AWS DMS utilisation du SDK AWS CLI ou d'un SDK, vous vous assurez que toutes les AWS DMS activités

Nouvelle fonction ou amélioration	Description
VPN ou connexion. AWS Direct Connect	restent invisibles pour l'Internet public. Pour plus d'informations, consultez Sécurité de l'infrastructure dans AWS Database Migration Service. **Note** Cette fonctionnalité est disponible avec toutes les versions de AWS DMS moteur prises en charge.
Partitionnement de dossiers basé sur la date de CDC en utilisant Amazon S3 en tant que cible	AWS DMS prend désormais en charge le partitionnement de dossiers basé sur la date lors de la réplication de données en utilisant S3 comme cible. Pour plus d'informations, consultez Utilisation du partitionnement de dossiers basé sur la date.

Les problèmes résolus dans la AWS DMS version 3.4.2 sont les suivants :

- Ajout d'une option `STATUPDATE` lors de l'exécution d'une migration utilisant Redshift en tant que cible.
- Amélioration des tâches de validation via l'introduction d'un nouveau paramètre. `ValidQueryCdcDelaySecond` retarde la première requête de validation sur les points de terminaison sources et cibles afin de réduire les conflits de ressources lorsque la latence de migration est élevée.
- Correction d'un problème en AWS DMS raison duquel le démarrage des tâches de validation prenait du temps.
- Problème résolu : des enregistrements vides étaient générés lors du démarrage ou de l'arrêt de tâches de réplication utilisant S3 en tant que cible.
- Problème résolu : les tâches se bloquaient une fois le chargement complet terminé.
- Problème résolu : les tâches se bloquaient lorsqu'une table source contenait des erreurs de données lors de l'utilisation de S3 en tant que source.
- Problème résolu : les tâches se bloquaient à leur démarrage lorsque le compte d'utilisateur du point de terminaison source était désactivé.

- Problème résolu : les tâches s'interrompaient lors de l'utilisation de PostgreSQL en tant que source avec `REPLICA IDENTITY FULL`.
- Problème résolu : les tâches manquaient des transactions lors de l'utilisation de PostgreSQL en tant que source avec le plug-in pglogical.
- Correction d'un problème qui AWS DMS empêchait de supprimer les fichiers source compressés lors de l'utilisation de Redshift comme cible.
- Problème résolu : les tâches de validation signalaient de faux négatifs lors de l'utilisation de MySQL à la fois en tant que source et cible avec le type de données `BIGINT UNSIGNED`.
- Problème résolu : les tâches de validation signalaient des faux positifs lors de l'utilisation de SQL Server en tant que source avec une colonne de clé primaire de type CHAR.
- Correction d'un problème en raison duquel les objets cibles AWS DMS ne sont pas effacés lors `start-replication` de l'utilisation pour démarrer des tâches de réplication en utilisant S3 comme cible.
- Plusieurs problèmes résolus relatifs à la validation des données lors de l'utilisation de Db2 en tant que source.
- Problème résolu : les tâches de validation se bloquaient lors de l'utilisation de SQL Server en tant que source avec la colonne VARCHAR comme clé primaire.
- Ajout de la prise en charge du type de données TIMESTAMP WITH TIMEZONE lors de l'utilisation de PostgreSQL en tant que source

AWS Notes de mise à jour de Database Migration Service 3.4.1 Bêta

Le tableau suivant présente les nouvelles fonctionnalités et améliorations introduites dans AWS DMS version 3.4.1 bêta.

Nouvelle fonction ou amélioration	Description
Nouvelle version de MongoDB	La version 4.0 de MongoDB est désormais prise en charge en tant que source.

Nouvelle fonction ou amélioration	Description
Prise en charge du protocole TLS 1.2 pour SQL Server	AWS DMS prend désormais en charge le protocole TLS 1.2 pour les points de terminaison SQL Server.

Les problèmes résolus dans la version bêta de la version AWS DMS 3.4.1 sont les suivants :

- Amélioration de la prise en charge d'Oracle 19c TDE.
- Amélioration de la prise en charge du jeu de caractères utf8mb4 et du type de données d'identité en utilisant Redshift en tant que cible.
- Amélioration de la gestion des échecs des tâches de réplication lors de l'utilisation de MySQL en tant que source et en l'absence de journal binaire.
- Amélioration de la prise en charge de la validation des données pour différents types de données et jeux de caractères.
- Amélioration de la gestion des valeurs null avec un nouveau paramètre de point de terminaison `IncludeNullAndEmpty` lors de l'utilisation de Kinesis et de Kafka en tant que cible.
- Amélioration de la gestion et de la journalisation des erreurs lors de l'utilisation de Kafka en tant que cible.
- Amélioration du décalage horaire DST lors de l'utilisation de SQL Server en tant que source.
- Problème résolu : les tâches de réplication tentaient de créer des tables existantes pour Oracle en tant que cible.
- Problème résolu : les tâches de réplication se bloquaient lorsque la connexion à la base de données était interrompue lors de l'utilisation d'Oracle en tant que source.
- Problème résolu : les tâches de réplication ne parvenaient pas à détecter le nouveau principal et à s'y reconnecter lors de l'utilisation de SQL Server en tant que source avec le paramètre AlwaysON.
- Problème résolu : les tâches de réplication n'ajoutaient pas un `"D"` pour la colonne `"0P"` dans certaines conditions pour S3 en tant que cible.

AWS Notes de mise à jour de la version bêta de Database Migration Service 3.4.0

Le tableau suivant présente les nouvelles fonctionnalités et améliorations introduites dans AWS DMS version 3.4.0.

Nouvelle fonction ou amélioration	Description
Nouvelle version de MySQL	AWS DMS supporte désormais la version 8.0 de MySQL en tant que source, sauf lorsque la charge utile de la transaction est compressée.
Prise en charge du protocole TLS 1.2 pour MySQL	AWS DMS supporte désormais le protocole TLS 1.2 pour les points de terminaison MySQL.
Nouvelle version de MariaDB	AWS DMS prend désormais en charge la version 10.3.13 de MariaDB en tant que source.
Absence SysAdmin d'accès aux sources Microsoft SQL Server autogérées	AWS DMS prend désormais en charge l'accès des SysAdmin non-utilisateurs aux points de terminaison source SQL Server sur site et hébergés par EC2. **Note** Cette fonctionnalité est actuellement en version bêta. Si vous souhaitez l'essayer, contactez le AWS support pour plus d'informations.
Tâches CDC et tables sources Oracle créées à l'aide de `CREATE TABLE AS`	AWS DMS prend désormais en charge à la fois les tâches à chargement complet, les tâches CDC et CDC uniquement exécutées sur les tables sources Oracle créées à l'aide de l'instruction. `CREATE TABLE AS`

Les problèmes résolus dans la AWS DMS version 3.4.0 sont les suivants :

- Amélioration des évaluations des tâches de prémigration. Pour plus d'informations, consultez Activation et utilisation des évaluations de prémigration pour une tâche.
- Amélioration de la validation des données pour les types de données float, real et double.
- Amélioration d'Amazon Redshift en tant que cible avec une meilleure gestion de l'erreur : « The specified key does not exist. »
- Prend en charge les paramètres des tâches de chargement CDC multithread`ParallelApplyThreads`, y compris`ParallelApplyBufferSize`, et`ParallelApplyQueuesPerThread`, pour Amazon OpenSearch Service (OpenSearch Service) en tant que cible.
- OpenSearch Service en tant que cible amélioré en soutenant son utilisation de clés primaires composites.
- Problème résolu : le test de connexion échouait lors de l'utilisation de PostgreSQL en tant que source, si le mot de passe contenait des caractères spéciaux.
- Correction d'un problème lié à l'utilisation de SQL Server en tant que source lorsque certaines colonnes `VARCHAR` étaient tronquées.
- Correction d'un problème qui AWS DMS empêchait de fermer les transactions ouvertes lors de l'utilisation d'Amazon RDS SQL Server comme source. Cela pouvait entraîner une perte de données si le paramètre d'intervalle d'interrogation était mal défini. Pour plus d'informations sur la configuration d'une valeur d'intervalle d'interrogation recommandée, consultez Utilisation d'une base de données Microsoft SQL Server comme source pour AWS DMS.
- Correction d'un problème lié à Oracle Standby en tant que source : les tâches de CDC s'arrêtaient de façon inattendue lors de l'utilisation de Binary Reader.
- Correction d'un problème lié à IBM DB2 pour LUW : la tâche échouait avec le message « The Numeric literal 0 is not valid because its value is out of range. »
- Correction d'un problème de migration de PostgreSQL vers PostgreSQL lorsqu'une nouvelle colonne était ajoutée dans la source PostgreSQL et que la colonne était créée avec un type de données différent de celui pour lequel la colonne avait été initialement créée dans la source.
- Correction d'un problème lié à une source MySQL : la tâche de migration s'arrêtait de façon inattendue lorsqu'elle ne parvenait pas à récupérer les journaux binaires.
- Correction d'un problème lié à une cible Oracle quand `BatchApply` était utilisé.
- Correction d'un problème lié à MySQL et MariaDB lors de la migration du type de données `TIME`.
- Correction d'un problème lié à une source IBM DB2 LUW : la migration de tables contenant des objets LOB échouait lorsque les tables ne possédaient pas de clé primaire ou de clé unique.

AWS Notes de mise à jour du Database Migration Service 3.3.4

Les problèmes résolus dans la version AWS DMS 3.3.4 sont les suivants :

- Problème résolu : les transactions étaient supprimées ou dupliquées lors de l'utilisation de PostgreSQL en tant que source.
- Amélioration de la prise en charge de l'utilisation du signe dollar ($) dans les noms de schéma.
- Problème résolu : les instances de réplication ne fermaient pas les transactions ouvertes lors de l'utilisation de RDS SQL Server en tant que source.
- Problème résolu : le test de connexion échouait lors de l'utilisation de PostgreSQL en tant que source, si le mot de passe contenait des caractères spéciaux.
- Amélioration d'Amazon Redshift en tant que cible avec une meilleure gestion de l'erreur : « The specified key does not exist. »
- Amélioration de la prise en charge de la validation des données pour différents types de données et jeux de caractères.
- Problème résolu : les tâches de réplication tentaient de créer des tables existantes pour Oracle en tant que cible.
- Problème résolu : les tâches de réplication n'ajoutaient pas un `"D"` pour la colonne `"OP"` dans certaines conditions pour Amazon S3 en tant que cible.

AWS Notes de mise à jour du Database Migration Service 3.3.3

Le tableau suivant présente les nouvelles fonctionnalités et améliorations introduites dans AWS DMS version 3.3.3.

Nouvelle fonction ou amélioration	Description
Nouvelle version de PostgreSQL	PostgreSQL version 12 est désormais pris en charge en tant que source et cible.
Support pour la clé primaire composite avec Amazon OpenSearch Service comme cible	À partir de AWS DMS 3.3.3, l'utilisation d'une clé primaire composite est prise en charge par les cibles du OpenSearch service.

Nouvelle fonction ou amélioration	Description
Prise en charge des types de données étendus Oracle	Les types de données étendus Oracle pour la source et les cibles Oracle sont désormais pris en charge.
Augmentation du nombre de AWS DMS ressources par compte	La limite du nombre de AWS DMS ressources que vous pouvez créer a été augmentée. Pour plus d'informations, consultez Quotas pour AWS Database Migration Service.

Les problèmes résolus dans la version AWS DMS 3.3.3 sont les suivants :

- Problème résolu : une tâche s'interrompait en utilisant une instruction de mise à jour spécifique avec une application parallèle dans Amazon Kinesis.
- Problème résolu : une tâche s'interrompait sur l'instruction ALTER TABLE avec Amazon S3 en tant que cible.
- Correction d'un problème à cause duquel des valeurs des colonnes de polygone étaient tronquées lors de l'utilisation de Microsoft SQL Server en tant que source.
- Correction d'un problème sur le convertisseur Unicode de JA16SJISTILDE et JA16EUCTILDE lors de l'utilisation d'Oracle en tant que source.
- Correction d'un problème qui empêchait la migration des colonnes MEDIUMTEXT et LONGTEXT de MySQL vers S3 au format CSV (valeur séparée par des virgules).
- Correction d'un problème provoquant la transformation des colonnes booléennes en types incorrects avec la sortie Apache Parquet.
- Correction d'un problème avec les colonnes varchar étendues dans Oracle.
- Correction d'un problème qui faisait échouer les tâches de validation des données en raison de certaines combinaisons d'horodatage.
- Correction d'un problème avec la réplication du langage de définition de données (DDL) Sybase.
- Correction d'un problème impliquant une source Oracle Real Application Clusters (RAC) qui plantait avec Oracle Binary Reader.
- Correction d'un problème lors de la validation des cibles Oracle avec la casse des noms de schéma.
- Correction d'un problème lors de la validation des versions 9.7 et 10 d'IBM Db2.

- Correction d'un problème lié à une tâche qui ne s'arrêtait pas deux fois avec `StopTaskCachedChangesApplied` et `StopTaskCachedChangesNotApplied` activées.

Historique du document

Le tableau suivant décrit les modifications majeures apportées au Guide de l'utilisateur AWS Database Migration Service depuis janvier 2018.

Vous pouvez vous abonner à un flux RSS afin de recevoir les notifications des mises à jour de cette documentation. Pour plus de détails sur les versions d'AWS DMS, consultez AWS Notes de mise à jour du DMS.

Modification	Description	Date
AWS DMS a ajouté la prise en charge de RDS IBM DB2 comme cible	AWS DMS prend désormais en charge l'utilisation d'Amazon RDS IBM DB2 comme cible.	4 décembre 2023
AWS DMS a ajouté la prise en charge de Timestream comme cible.	AWS DMS prend désormais en charge Timestream comme cible.	17 novembre 2023
AWS DMS a ajouté la prise en charge de la validation des données dans les cibles Redshift	AWS DMS prend désormais en charge la validation des données dans les cibles Redshift.	14 novembre 2023
AWS DMS a ajouté la prise en charge de quatre nouveaux types de points de terminaison	AWS DMS prend désormais en charge l'utilisation de Microsoft Azure Database for PostgreSQL, Microsoft Azure Database for MySQL, OCI MySQL Heatwave et Google Cloud for PostgreSQL en tant que source.	26 octobre 2023
AWS DMS a ajouté la prise en charge d'un nouveau rôle lié à un service AWS	AWS DMS prend désormais en charge le rôle lié à un service AWS `AWSServiceRoleForDMSServerl`	22 mai 2023

	`ess` qui permet à AWS DMS de créer et gérer des ressources en votre nom, par exemple en publiant des points de données de métriques sur Amazon CloudWatch.	
AWS DMS a ajouté la prise en charge d'une nouvelle politique gérée AWS	AWS DMS prend désormais en charge la politique gérée AWS qui permet de publier des journaux de réplication sans serveur sur CloudWatch Logs.	22 mai 2023
AWS DMS a ajouté la prise en charge d'une nouvelle politique gérée AWS	AWS DMS prend désormais en charge la politique gérée AWS qui permet de publier des points de données de métriques dans Amazon CloudWatch. AWS DMS a également commencé à suivre les modifications pour ses politiques gérées AWS.	06 mars 2023
Prise en charge des points de terminaison sources et cibles de VPC	AWS DMS prend désormais en charge les points de terminaison de cloud privé virtuel (VPC) en tant que sources et cibles. AWS DMS peut désormais se connecter à n'importe quel service AWS doté de points de terminais on de VPC lorsque des routes explicitement définies vers ces services sont définies dans leur VPC AWS DMS.	30 juin 2022

Prise en charge des réplicas de lecture SQL Server en tant que source	AWS DMS prend désormais en charge les réplicas de lecture SQL Server en tant que source. Avec AWS DMS, vous pouvez désormais effectuer des migrations en direct depuis un réplica de lecture SQL Server vers n'importe quelle cible AWS DMS prise en charge.	30 juin 2022
Prise en charge des bases de données IBM Db2 z/OS en tant que source pour le chargement complet uniquement	AWS DMS prend désormais en charge les bases de données IBM Db2 z/OS en tant que source. Avec AWS DMS, vous pouvez désormais effectuer des migrations en direct depuis les mainframe s Db2 vers n'importe quelle cible prise en charge par AWS DMS.	30 juin 2022
Prise en charge des événements DMS d'EventBr idge	AWS DMS prend en charge la gestion des abonnements aux événements EventBridge à l'aide des événements DMS.	30 juin 2022
Prise en charge de Babelfish en tant que cible	AWS DMS prend désormais en charge Babelfish en tant que cible. Avec AWS DMS, vous pouvez désormais migrer des données en direct depuis n'importe quelle source prise en charge par AWS DMS vers une cible Babelfish, avec un temps d'arrêt minimal.	30 juin 2022

Prise en charge d'Aurora sans serveur v2 en tant que cible	AWS DMS prend désormais en charge Aurora sans serveur v2 en tant que cible. Avec AWS DMS, vous pouvez désormais effectuer des migrations en direct vers Aurora sans serveur v2.	30 juin 2022
Didacticiel de mise en route	Mise à jour du didacticiel de mise en route pour AWS DMS. Le didacticiel utilise une base de données MySQL en tant que source et une base de données PostgreSQL en tant que cible.	20 mai 2021
Prise en charge d'Amazon Neptune en tant que cible	Ajout de la prise en charge d'Amazon Neptune en tant que cible pour la migration des données.	1er juin 2020
Prise en charge d'Apache Kafka en tant que cible	Ajout de la prise en charge d'Apache Kafka comme cible pour la migration des données.	20 mars 2020
Mise à jour du contenu de sécurité	Contenu de sécurité mis à jour et standardisé en réponse aux demandes des clients.	20 décembre 2019
Migration avec AWS Snowball Edge	Ajout de la prise en charge de l'utilisation de AWS Snowball Edge pour migrer des bases de données volumineuses.	24 janvier 2019

Prise en charge d'Amazon DocumentDB (compatible avec MongoDB) en tant que cible	Ajout de la prise en charge d'Amazon DocumentDB (compatible avec MongoDB) en tant que cible pour la migration des données.	9 janvier 2019
Prise en charge d'Amazon OpenSearch Service et Amazon Kinesis Data Streams en tant que cibles	Ajout de la prise en charge d'OpenSearch Service et Kinesis Data Streams en tant que cibles pour la migration de données.	15 novembre 2018
Prise en charge du point de départ natif CDC	Ajout de la prise en charge des points de départ natifs lors de l'utilisation de la capture des données modifiées (CDC).	28 juin 2018
Prise en charge de DB2 LUW	Ajout de la prise en charge de IBM Db2 LUW comme source de la migration des données.	26 avril 2018
Prise en charge de SQL Server en tant que cible	Ajout de la prise en charge d'Amazon RDS for Microsoft SQL Server comme source.	6 février 2018

Glossaire AWS

Pour connaître la terminologie la plus récente d'AWS, consultez le Glossaire AWS dans la Référence Glossaire AWS.

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.