



Panduan Pengguna

AWS Layanan Migrasi Database



AWS Layanan Migrasi Database: Panduan Pengguna

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara para pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan milik dari masing-masing pemilik, yang mungkin atau mungkin tidak berafiliasi dengan, terhubung ke, atau disponsori oleh Amazon.

Table of Contents

Apa itu AWS Database Migration Service?	1
Tugas migrasi yang dijalankan oleh AWS DMS	2
Bagaimana cara AWS DMS kerja	4
Tampilan tingkat tinggi AWS DMS	4
Komponen-komponen	6
Sumber	13
Sumber untuk migrasi data	13
Sumber untuk DMS Fleet Advisor	16
Sumber untuk Konversi Skema DMS	17
Sumber untuk migrasi data homogen DMS	17
Target	18
Target migrasi data	18
Target untuk DMS Fleet Advisor	20
Target untuk Konversi Skema DMS	21
Target untuk migrasi data homogen DMS	21
Nama sumber daya Amazon	22
Dengan AWS layanan lain	24
Support untuk AWS CloudFormation	25
Memulai	26
Mengatur	26
Daftar Akun AWS	26
Membuat pengguna administratif	27
Prasyarat	28
Buat VPC	29
Buat grup parameter Amazon RDS	30
Membuat basis data Amazon RDS sumber Anda	32
Membuat basis data Amazon RDS	33
Membuat klien Amazon EC2	34
Mengisi basis data sumber Anda	35
Migrasi skema	37
Replikasi	39
Langkah 1: Membuat instans replikasi	39
Langkah 2: Tentukan titik akhir sumber dan target	41
Langkah 3: Buat tugas dan migrasi data	43

Langkah 4: Replikasi pengujian	45
Langkah 5: Bersihkan Sumber Daya AWS DMS	47
Sumber daya tambahan	49
Menemukan database untuk migrasi	50
Wilayah AWS yang Didukung	51
Mulai	53
Menyiapkan	53
Membuat sumber daya yang dibutuhkan	54
Membuat pengguna database	63
Pengumpul data	70
Izin	71
Membuat pengumpul data	71
Memasang pengumpul data	74
Menemukan server OS dan database	77
Mengelola objek yang dipantau	81
Menggunakan SSL	83
Mengumpulkan data	84
Memecahkan masalah	90
Inventaris	92
Menggunakan inventaris database	93
Menggunakan inventaris skema	94
Rekomendasi target	96
Contoh target	97
Bagaimana DMS Fleet Advisor menentukan spesifikasi target?	97
Menghasilkan rekomendasi target	98
Rincian rekomendasi	100
Mengekspor rekomendasi target	102
Batasan migrasi	103
Memecahkan masalah	122
Batasan	123
Mengonversi skema database	125
Didukung Wilayah AWS	126
Fitur	127
Batasan	128
Memulai	129
Prasyarat	130

Langkah 1: Buat profil instance	134
Langkah 2: Konfigurasi penyedia data	135
Langkah 3: Buat proyek migrasi	136
Langkah 4: Buat laporan penilaian	137
Langkah 5: Konversi kode sumber Anda	137
Langkah 6: Terapkan kode yang dikonversi	138
Langkah 7: Bersihkan dan pecahkan masalah	138
Menyiapkan jaringan	140
Konfigurasi VPC tunggal	140
Beberapa konfigurasi VPC	140
Menggunakan AWS Direct Connect atau VPN	141
Menggunakan koneksi internet	141
Menggunakan lingkungan tanpa gateway Internet	142
Membuat penyedia data sumber	142
Menggunakan SQL Server sebagai sumber	143
Menggunakan Oracle sebagai sumber	145
Menggunakan Oracle Data Warehouse sebagai sumber	146
Menggunakan PostgreSQL sebagai sumber	148
Menggunakan MySQL sebagai sumber	149
Membuat penyedia data target	150
Menggunakan MySQL sebagai target	150
Menggunakan PostgreSQL sebagai target	152
Menggunakan Amazon Redshift sebagai target	153
Mengelola proyek migrasi	154
Menentukan pengaturan proyek migrasi	154
Laporan penilaian migrasi database	155
Membuat laporan penilaian	156
Melihat laporan penilaian Anda	157
Menyimpan laporan penilaian	158
Konversi skema	160
Menyiapkan aturan transformasi	161
Mengonversi skema database Anda	163
Menentukan pengaturan konversi skema	167
Menyegarkan skema database Anda	172
Menyimpan dan menerapkan skema Anda	172
Menggunakan paket ekstensi	174

Migrasi data homogen	176
Wilayah AWS yang Didukung	177
Fitur	178
Batasan	178
Gambaran Umum	179
Pengaturan	180
Membuat sumber daya IAM	180
Menyiapkan jaringan	184
Membuat penyedia data sumber	188
Menggunakan MySQL atau MariaDB sebagai sumber	189
Menggunakan PostgreSQL sebagai sumber	193
Menggunakan MongoDB atau Amazon DocumentDB sebagai sumber	196
Membuat penyedia data target	199
Menggunakan MySQL atau MariaDB sebagai target	200
Menggunakan PostgreSQL sebagai target	202
Menggunakan Amazon DocumentDB sebagai target	203
Migrasi data	204
Membuat migrasi data	204
Aturan seleksi	207
Mengelola migrasi data	209
Memantau migrasi data	211
Status migrasi	213
Migrasi data dari MySQL	214
Migrasi data dari PostgreSQL	215
Migrasi data dari MongoDB	217
Pemecahan Masalah	218
Membuat migrasi data	219
Memulai migrasi data	219
Masalah koneksi	219
Tampilan bermigrasi sebagai tabel di PostgreSQL	220
Bekerja dengan proyek migrasi	221
Membuat grup subnet	222
Membuat profil instance	222
Membuat penyedia data	224
Membuat proyek migrasi	226
Mengelola proyek migrasi	227

Praktik terbaik	229
Perencanaan migrasi untuk AWS Database Migration Service	229
Konversi skema	231
Meninjau dokumentasi AWS DMS	231
Menjalankan bukti konsep	232
Meningkatkan performa	232
Menggunakan server nama on-premise Anda sendiri	238
Menggunakan Amazon Route 53 Resolver dengan AWS DMS	239
Melakukan migrasi objek biner large (LOB)	240
Menggunakan mode LOB terbatas	240
Peningkatan performa LOB	241
Meningkatkan performa ketika melakukan migrasi tabel-tabel large menggunakan filter baris ..	244
Replikasi yang sedang berlangsung	245
Mengurangi beban pada basis data sumber Anda	246
Mengurangi kemacetan pada basis data target Anda	246
Menggunakan validasi data	246
Pemantauan metrik	247
Tindakan	248
Menggunakan cara dengan mencatat tugas	248
Memecahkan masalah replikasi dengan Time Travel	248
Mengubah pengguna dan skema untuk target Oracle	249
Mengubah tempat penyimpanan tabel dan indeks untuk target Oracle	250
Pemutakhiran instans replikasi	251
Memahami biaya migrasi	251
Bekerja dengan Tanpa AWS DMS Server	252
Komponen tanpa server DMS	253
Versi Mesin yang Didukung	256
Membuat replikasi tanpa server	257
Memodifikasi replikasi AWS DMS tanpa server	259
Konfigurasi Komputasi	263
Memahami penskalaan otomatis tanpa server AWS DMS	264
Memantau AWS DMS replikasi tanpa server	265
Batasan tanpa server	270
Bekerja dengan instans replikasi	272
Memilih tipe instans replikasi	277
Menentukan kelas instans yang digunakan	282

Mode tak terbatas instans yang dapat dilonjakkan	283
Pengukuran instans replikasi	284
Faktor yang perlu dipertimbangkan	284
Masalah umum	286
Praktik terbaik	286
Versi mesin replikasi	287
Meningkatkan versi mesin menggunakan konsol	287
Memutakhirkan versi mesin menggunakan AWS CLI	288
Instans replikasi publik dan privat	289
Alamat IP dan jenis jaringan	289
Menyiapkan jaringan untuk instans replikasi	291
Konfigurasi jaringan untuk migrasi basis data	292
Membuat grup subnet replikasi	300
Menyelesaikan titik akhir domain menggunakan DNS	301
Mengatur kunci enkripsi	302
Membuat instans replikasi	303
Mengubah instans replikasi	308
Mem-boot ulang instans replikasi	314
Menghapus instans replikasi	317
Jendela pemeliharaan DMS	319
Pengaruh pemeliharaan pada tugas migrasi yang ada	319
Mengubah pengaturan jendela pemeliharaan	319
Titik akhir	322
Membuat titik akhir sumber dan target	322
Sumber untuk migrasi data	327
Menggunakan Oracle sebagai sumber	328
Menggunakan SQL Server sebagai sumber	399
Menggunakan basis data Azure SQL sebagai sumber	429
Menggunakan Azure SQL Managed Instance sebagai sumber	429
Menggunakan Azure Database untuk PostgreSQL sebagai sumber	430
Menggunakan Azure Database untuk MySQL sebagai sumber	431
Menggunakan OCI MySQL Heatwave sebagai sumber	432
Menggunakan Google Cloud untuk MySQL sebagai sumber	433
Menggunakan Google Cloud untuk PostgreSQL sebagai sumber	433
Menggunakan PostgreSQL sebagai sumber	435
Menggunakan MySQL sebagai sumber	473

Menggunakan SAP ASE sebagai sumber	486
Menggunakan MongoDB sebagai sumber	495
Menggunakan Amazon DocumentDB sebagai sumber	513
Menggunakan Amazon S3 sebagai sumber	530
Menggunakan IBM Db2 LUW sebagai sumber	542
Menggunakan IBM Db2 untuk z/OS sebagai sumber	550
Target migrasi data	590
Menggunakan Oracle sebagai target	591
Menggunakan SQL Server sebagai target	603
Menggunakan PostgreSQL sebagai target	609
Menggunakan MySQL sebagai target	622
Menggunakan Amazon Redshift sebagai target	631
Menggunakan SAP ASE sebagai target	657
Menggunakan Amazon S3 sebagai target	660
Menggunakan Amazon DynamoDB sebagai target	711
Menggunakan Amazon Kinesis Data Streams sebagai target	732
Menggunakan Apache Kafka sebagai target	751
Menggunakan OpenSearch sebagai target	779
Menggunakan Amazon DocumentDB sebagai target	785
Menggunakan Amazon Neptune sebagai target	798
Menggunakan Redis sebagai target	815
Menggunakan Babelfish sebagai target	822
Menggunakan Amazon Timestream sebagai target	830
Menggunakan Db2 sebagai target	842
Titik akhir VPC untuk migrasi data	843
Siapa yang terpengaruh saat bermigrasi ke AWS DMS versi 3.4.7 dan lebih tinggi?	844
Siapa yang tidak terpengaruh saat bermigrasi ke AWS DMS versi 3.4.7 dan lebih tinggi?	844
Mempersiapkan migrasi ke AWS DMS versi 3.4.7 dan lebih tinggi	845
Pernyataan DDL yang didukung	846
Tugas	848
Membuat tugas	852
Pengaturan tugas	860
Pengaturan dukungan LOB	913
Membuat beberapa tugas	915
Tugas replikasi terus-menerus	916
Replikasi mulai dari titik awal CDC	918

Melakukan replikasi dua arah	923
Mengubah tugas	927
Memindahkan tugas	927
Memuat ulang tabel selama mengerjakan tugas	929
AWS Management Console	929
Pemetaan tabel	931
Menentukan pemilihan tabel dan transformasi aturan dari konsol	931
Menentukan pemilihan tabel dan transformasi aturan menggunakan JSON	935
Aturan seleksi dan tindakan	937
Wildcard dalam pemetaan tabel	944
Aturan dan tindakan transformasi	945
Menggunakan ekspresi aturan transformasi untuk menentukan isi kolom	968
Tabel dan koleksi pengaturan aturan dan operasi	983
Menggunakan filter sumber	1015
Menerapkan filter	1016
Penyaringan berdasarkan waktu dan tanggal	1022
Mengaktifkan dan bekerja dengan penilaian perdana	1023
Prasyarat	1024
Menentukan, memulai, dan melihat penilaian berjalan	1027
Penilaian individu	1031
Memulai dan melihat penilaian tipe data	1052
Penilaian pemecahan masalah berjalan	1056
Menentukan data tambahan	1057
Pemantauan Tugas	1058
Status Tugas	1060
Kondisi tabel selama tugas	1063
Pemantauan tugas replikasi menggunakan Amazon CloudWatch	1064
Metrik AWS Database Migration Service	1066
Metrik instans replikasi	1069
Metrik tugas replikasi	1072
Melihat dan mengelola AWSMencatat DMS	1075
Mencatat panggilan API AWS DMS dengan AWS CloudTrail	1077
AWS DMSinformasi di CloudTrail	1077
Memahami entri file log AWS DMS	1078
Endit log koneksi	1082
Jenis Objek	1082

Contoh Pencatatan	1084
Keterbatasan:	1085
bekerja dengan EventBridge acara	1086
menggunakan Amazon EventBridge aturan acara untuk AWS DMS	1087
AWS DMS kategori tindakan dan tindakan	1088
ReplicationInstance tindakan	1088
ReplicationTask tindakan	1092
Pesan tindakan replikasi	1094
Bekerja dengan acara Amazon SNS	1096
Memindahkan langganan acara ke AmazonEventBridge	1096
Bekerja dengan acara dan notifikasi Amazon SNS	1097
AWS Kategori acara DMS dan pesan acara untuk pemberitahuan SNS	1098
Berlangganan AWS Pemberitahuan acara DMS menggunakan SNS	1102
Menggunakan AWS Management Console	1103
Memvalidasi kebijakan akses topik SNS Anda	1105
Validasi data	1107
Statistik tugas replikasi	1108
Statistik tugas replikasi dengan Amazon CloudWatch	1111
Memvalidasi ulang tabel selama pengerjaan tugas	1112
AWS Management Console	1112
Menggunakan editor JSON untuk memodifikasi aturan validasi	1112
Tugas hanya validasi	1113
Validasi beban penuh saja	1114
Hanya validasi CDC	1114
Validasi hanya kasus penggunaan	1114
Pemecahan Masalah	1115
Kinerja Validasi Pergeseran Merah	1117
Batasan	1118
Validasi S3	1119
Prasyarat	1120
Izin	1120
Batasan	1122
Tugas hanya validasi	1123
Penandaan sumber daya	1124
API	1126
Keamanan	1128

Perlindungan data	1131
Enkripsi data	1131
Privasi lalu lintas kerja internet	1132
Perlindungan data di DMS Fleet Advisor	1132
Pengelolaan identitas dan akses	1134
Audiens	1134
Mengautentikasi dengan identitas	1135
Mengelola akses menggunakan kebijakan	1138
Bagaimana AWS Database Migration Service bekerja dengan IAM	1141
Contoh kebijakan berbasis identitas	1148
Contoh kebijakan berbasis sumber daya	1157
Menggunakan rahasia untuk mengakses sumber daya	1162
Menggunakan peran terkait layanan	1172
Memecahkan masalah	1179
Izin IAM yang diperlukan	1182
IAM role untuk CLI dan API	1187
Pencegahan confused deputy lintas layanan	1193
AWSKebijakan yang dikelola	1196
Validasi kepatuhan	1203
Ketahanan	1205
Keamanan infrastruktur	1206
Kontrol akses detail	1209
Menggunakan nama sumber daya untuk mengontrol akses	1209
Menggunakan tag untuk mengontrol akses	1212
Mengatur kunci enkripsi	1220
Keamanan jaringan	1223
Menggunakan SSL	1225
Keterbatasan menggunakan SSL dengan AWS DMS	1227
Mengelola sertifikat	1227
Mengaktifkan SSL untuk titik akhir yang kompatibel dengan MySQL, PostgreSQL, atau SQL Server	1228
Pengubahan kata sandi basis data	1231
Batasan	1232
Kuota sumber daya AWS Database Migration Service	1232
Memahami throttling permintaan API	1234
Pemecahan masalah dan dukungan diagnostik	1235

Tugas migrasi berjalan perlahan	1236
Bilah status tugas tidak bergerak	1237
Tugas selesai tapi tidak ada yang dimigrasi	1237
Kunci asing dan indeks sekunder hilang	1237
AWS DMStidak membuat CloudWatch log	1238
Masalah terjadi saat terhubung ke Amazon RDS	1238
Pesan kesalahan: koneksi string thread salah: nilai thread salah 0	1239
Masalah jaringan terjadi	1239
CDC tertahan setelah beban penuh	1240
Kesalahan pelanggaran kunci primer terjadi saat Anda memulai ulang tugas	1240
Beban awal skema gagal	1240
Tugas gagal dengan kesalahan yang tidak diketahui	1241
Pemulaian ulang tugas memuat tabel dari awal	1241
Jumlah tabel per tugas menyebabkan masalah	1241
Tugas gagal ketika kunci primer dibuat pada kolom LOB	1241
Catatan duplikat terjadi pada tabel target tanpa kunci primer	1241
Titik akhir sumber termasuk dalam rentang IP yang dicadangkan	1242
Stempel waktu kacau dalam pertanyaan Amazon Athena	1242
Pemecahan masalah dengan Oracle	1242
Menarik data dari tampilan	1243
Memigrasi LOB dari Oracle 12c	1243
Beralih antara Oracle LogMiner dan Binary Reader	1243
Kesalahan: Oracle CDC berhenti 122301 penghitung coba kembali oracle CDC terlampau.	1244
Secara otomatis menambahkan supplemental logging untuk titik akhir sumber Oracle	1245
Perubahan LOB tidak sedang ditangkap	1245
Kesalahan: ORA-12899: Nilai terlalu besar untuk kolom <i>nama-kolom</i>	1246
Tipe data ANGKA yang disalahartikan	1246
Catatan hilang selama beban penuh	1246
Kesalahan Tabel	1247
Kesalahan: Tidak dapat mengambil id tujuan log Redo yang diarsipkan Oracle	1247
Mengevaluasi kinerja baca Oracle redo atau arsip log	1247
Pemecahan masalah dengan MySQL	1249
Tugas CDC gagal untuk titik akhir instans DB Amazon RDS karena log biner dinonaktifkan	1250
Koneksi ke instans MySQL target terputus selama pengerjaan tugas	1250

Menambahkan autocommit ke titik akhir yang kompatibel dengan MySQL	1251
Menonaktifkan kunci asing pada titik akhir yang kompatibel dengan MySQL target	1251
Karakter diganti dengan tanda tanya	1252
Entri log "Peristiwa buruk"	1252
Mengubah tangkapan data dengan MySQL 5.5	1252
Meningkatkan retensi log biner untuk instans DB Amazon RDS	1252
Pesan log: Beberapa perubahan dari basis data sumber tidak memiliki dampak ketika diterapkan ke basis data target.	1253
Kesalahan: Pengidentifikasi terlalu panjang	1253
Kesalahan: Set karakter yang tidak didukung menyebabkan konversi data bidang gagal ...	1253
Kesalahan: Codepage 1252 ke UTF8 [120112] konversi data bidang gagal	1254
Indeks, Kunci Asing, atau Pembaruan atau Penghapusan Cascade Tidak Dimigrasi	1255
Pemecahan masalah dengan PostgreSQL	1256
Jenis data JSON dipotong	1257
Kolom dari tipe data yang ditetapkan pengguna tidak dimigrasi dengan benar	1258
Kesalahan: Tidak ada skema yang dipilih untuk dibuat di dalamnya	1258
Penghapusan dan pembaruan ke tabel tidak direplikasi menggunakan CDC	1258
Pernyataan memotong tidak disebarakan	1258
Mencegah PostgreSQL dari menangkap DDL	1259
Memilih skema di mana objek basis data untuk menangkap DDL dibuat	1259
Tabel Oracle hilang setelah bermigrasi ke PostgreSQL	1259
ReplicationSlotDiskUsage meningkat dan restart_Isn berhenti bergerak maju selama transaksi panjang, seperti beban kerja ETL	1259
Tugas yang menggunakan tampilan sebagai sumber tidak memiliki baris yang disalin	1260
Pemecahan masalah Microsoft SQL Server	1260
Kesalahan menangkap perubahan untuk basis data SQL server	1261
Kolom identitas hilang	1261
Kesalahan: SQL Server tidak mendukung publikasi	1261
Perubahan tidak muncul di target Anda	1261
Tabel yang tidak seragam dipetakan di seluruh partisi	1262
Pemecahan masalah dengan Amazon Redshift	1263
Memuat ke sebuah klaster Amazon Redshift di Wilayah AWS yang berbeda	1263
Kesalahan: Relasi "awsdms_apply_exceptions" sudah ada	1263
Kesalahan dengan tabel yang namanya dimulai dengan "awsdms_changes"	1263
Melihat tabel dalam klaster dengan nama seperti dms.awsdms_changes000000000XXXX	1263
Izin yang diperlukan untuk bekerja dengan Amazon Redshift	1264

Pemecahan masalah dengan Amazon Aurora MySQL	1264
Kesalahan: SET KARAKTER bidang UTF8 diakhiri oleh ',' diapit oleh "" baris diakhiri oleh '\n'	1264
Memecahkan masalah dengan SAP ASE	1265
Kesalahan: Kolom LOB memiliki nilai NULL ketika sumber memiliki indeks unik komposit dengan nilai NULL	1265
Memecahkan masalah dengan IBM Db2	1265
Kesalahan: Lanjutkan dari stempel waktu tidak didukung Tugas	1265
Pemecahan masalah latensi	1266
Jenis latensi CDC	1266
Penyebab umum latensi CDC	1267
Memecahkan masalah latensi	1271
Bekerja dengan skrip dukungan diagnostik	1285
Skrip dukungan Oracle	1287
Skrip dukungan SQL Server	1290
Skrip dukungan yang kompatibel dengan MySQL	1316
Skrip dukungan PostgreSQL	1318
Bekerja dengan dukungan diagnostik AMI	1320
Luncurkan instans Amazon EC2 AWS DMS diagnostik baru	1321
Membuat IAM role	1322
Jalankan Tes Diagnostik	1323
Langkah Selanjutnya	1327
ID AMI menurut wilayah	1327
Referensi	1330
Tipe data AWS DMS	1330
Catatan rilis	1333
AWS DMS 3.5.2 catatan rilis	1334
AWS DMS 3.5.1 catatan rilis	1334
AWS DMS 3.5.0 Catatan rilis beta	1345
AWS DMS 3.4.7 catatan rilis	1351
AWS DMS 3.4.6 catatan rilis	1360
AWS DMS 3.4.5 catatan rilis	1367
AWS DMS 3.4.4 catatan rilis	1370
AWS DMS 3.4.3 catatan rilis	1372
AWS DMS 3.4.2 catatan rilis	1375
AWS DMS 3.4.1 catatan rilis	1376

AWS DMS 3.4.0 catatan rilis	1377
AWS DMS 3.3.4 catatan rilis	1379
AWS DMS 3.3.3 catatan rilis	1380
Riwayat dokumen	1382
AWSGlosarium	1387
.....	mccclxxxviii

Apa itu AWS Database Migration Service?

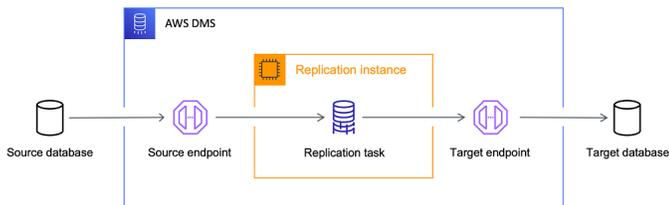
AWS Database Migration Service(AWS DMS) adalah layanan cloud yang memungkinkan untuk memigrasi database relasional, gudang data, database NoSQL, dan jenis penyimpanan data lainnya. Anda dapat menggunakan AWS DMS untuk memigrasikan data Anda ke AWS Cloud atau menggunakan kombinasi pengaturan cloud dan on-premises.

DenganAWS DMS, Anda dapat menemukan penyimpanan data sumber, mengonversi skema sumber, dan memigrasi data Anda.

- Untuk menemukan infrastruktur data sumber Anda, Anda dapat menggunakan DMS Fleet Advisor. Layanan ini mengumpulkan data dari database lokal dan server analitik, serta membuat inventaris server, database, dan skema yang dapat Anda migrasi ke Cloud. AWS
- Untuk bermigrasi ke mesin database yang berbeda, Anda dapat menggunakan Konversi Skema DMS. Layanan ini secara otomatis menilai dan mengubah skema sumber Anda ke mesin target baru. Atau, Anda dapat mengunduh AWS Schema Conversion Tool (AWS SCT) ke PC lokal Anda untuk mengonversi skema sumber Anda.
- Setelah Anda mengonversi skema sumber dan menerapkan kode yang dikonversi ke database target Anda, Anda dapat menggunakannya AWS DMS untuk memigrasi data Anda. Anda dapat melakukan migrasi satu kali atau mereplikasi perubahan yang sedang berlangsung untuk menjaga sumber dan target tetap sinkron. Karena AWS DMS adalah bagian dari AWS Cloud, yang Anda dapatkan adalah efisiensi biaya, kecepatan ke pasar, keamanan, dan fleksibilitas yang ditawarkan oleh layanan AWS.

Pada tingkat dasar, AWS DMS adalah sebuah server di AWS Cloud yang menjalankan perangkat lunak replikasi. Anda membuat sumber dan koneksi target untuk memberi tahu dari AWS DMS mana mengekstrak data dan dari mana memuatnya. Selanjutnya, Anda menjadwalkan tugas yang berjalan di server ini untuk memindahkan data Anda. AWS DMSmenciptakan tabel dan kunci primer terkait jika mereka tidak ada pada target. Anda dapat membuat tabel target sendiri jika Anda mau. Atau Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk membuat beberapa atau semua tabel target, indeks, views, pemicu, dan lain-lain.

Diagram berikut menggambarkan proses replikasi AWS DMS.



References

- AWS Wilayah yang mendukung AWS DMS - Untuk informasi tentang apa yang didukung AWS Wilayah AWS DMS, lihat [Bekerja dengan instance AWS DMS replikasi](#).
- Biaya migrasi database - Untuk informasi tentang biaya migrasi database, lihat [halaman AWS Database Migration Service harga](#).
- AWS DMS fitur dan manfaat — Untuk informasi tentang AWS DMS fitur dan manfaat, lihat [AWS Database Migration Service Fitur](#).
- Opsi database yang tersedia — Untuk mempelajari lebih lanjut tentang berbagai opsi database yang tersedia di Amazon Web Services, lihat [Memilih database yang tepat untuk organisasi Anda](#).

Tugas migrasi yang dijalankan oleh AWS DMS

AWS DMS mengambil alih banyak tugas yang sulit dan membosankan dalam proyek migrasi:

- Dalam solusi tradisional, Anda perlu melakukan analisis kapasitas, pengadaan perangkat keras dan perangkat lunak, menginstal dan mengelola sistem, dan menguji dan debug instalasi. AWS DMS secara otomatis mengelola deployment, pengelolaan, dan pemantauan semua perangkat keras dan perangkat lunak yang diperlukan untuk migrasi Anda. Migrasi Anda dapat aktif dan berjalan dalam beberapa menit setelah memulai proses konfigurasi AWS DMS.
- Dengan AWS DMS, Anda dapat menaikkan (atau menurunkan) skala sumber daya migrasi sesuai kebutuhan untuk mencocokkan beban kerja Anda yang sebenarnya. Misalnya, jika Anda memerlukan penyimpanan tambahan, Anda dapat dengan mudah meningkatkan penyimpanan yang dialokasikan dan memulai ulang migrasi dalam hitungan menit saja.
- AWS DMS menggunakan pay-as-you-go model. Anda hanya membayar untuk sumber daya AWS DMS saat Anda menggunakannya, dibandingkan dengan model lisensi tradisional dengan biaya pembelian di muka dan biaya pemeliharaan berkelanjutan.
- AWS DMS secara otomatis mengelola semua infrastruktur yang mendukung server migrasi Anda, termasuk perangkat keras dan perangkat lunak, patch perangkat lunak, dan pelaporan kesalahan.

- AWS DMS menyediakan failover otomatis. Jika server replikasi utama Anda gagal karena alasan apapun, server replikasi cadangan dapat mengambil alih dengan sedikit atau tanpa gangguan layanan.
- AWS DMS Fleet Advisor secara otomatis inventaris infrastruktur data Anda. Ini membuat laporan yang membantu Anda mengidentifikasi kandidat migrasi dan merencanakan migrasi Anda.
- AWS Konversi Skema DMS secara otomatis menilai kompleksitas migrasi Anda untuk penyedia data sumber Anda. Hal ini juga mengkonversi skema database dan objek kode ke format yang kompatibel dengan database target dan kemudian menerapkan kode dikonversi.
- AWS DMS dapat membantu Anda beralih ke mesin basis data modern, yang lebih hemat biaya dibandingkan yang Anda gunakan sekarang. Misalnya, AWS DMS bisa membantu Anda memanfaatkan layanan basis data terkelola Amazon Relational Database Service (Amazon RDS) atau Amazon Aurora. Atau dapat membantu Anda pindah ke layanan gudang data terkelola yang disediakan oleh Amazon Redshift, platform NoSQL seperti Amazon DynamoDB, atau platform penyimpanan berbiaya rendah seperti Amazon Simple Storage Service (Amazon S3). Sebaliknya, jika Anda ingin bermigrasi jauh dari infrastruktur lama tetapi terus menggunakan mesin basis data yang sama, AWS DMS juga mendukung proses tersebut.
- AWS DMS mendukung hampir semua mesin DBMS paling populer saat ini sebagai sumber endpoint. Untuk informasi selengkapnya, lihat [Sumber untuk migrasi data](#).
- AWS DMS menyediakan cakupan luas engine target yang tersedia. Untuk informasi selengkapnya, lihat [Target migrasi data](#).
- Anda dapat bermigrasi dari satu sumber data yang didukung ke lainnya. AWS DMS mendukung migrasi data sepenuhnya yang heterogen antara engine yang didukung.
- AWS DMS memastikan bahwa migrasi data Anda aman. Data at rest dienkripsi dengan enkripsi AWS Key Management Service (AWS KMS). Selama migrasi, Anda dapat menggunakan Secure Socket Layers (SSL) untuk mengenkripsi data in-flight Anda saat berpindah dari sumber ke target.

Cara kerja AWS Database Migration Service

AWS Database Migration Service (AWS DMS) adalah layanan web yang dapat Anda gunakan untuk memigrasikan data dari penyimpanan data sumber ke penyimpanan data target. Kedua penyimpanan data ini disebut titik akhir. Anda dapat melakukan migrasi antara titik akhir sumber dan target yang menggunakan mesin basis data yang sama, seperti dari basis data Oracle ke basis data Oracle. Anda juga dapat melakukan migrasi antara titik sumber sumber dan target yang menggunakan mesin basis data yang berbeda, seperti dari basis data Oracle ke basis data PostgreSQL. Satu-satunya persyaratan untuk digunakan AWS DMS adalah bahwa salah satu titik akhir Anda harus pada AWS layanan. Anda tidak dapat menggunakan AWS DMS untuk bermigrasi dari database lokal ke database lokal lainnya.

Untuk informasi tentang biaya migrasi basis data, lihat [AWS Database Migration Service Halaman harga](#).

Gunakan topik berikut untuk lebih memahami AWS DMS.

Topik

- [Tampilan tingkat tinggi AWS DMS](#)
- [Komponen AWS DMS](#)
- [Sumber untuk AWS DMS](#)
- [Target untuk AWS DMS](#)
- [Membangun Nama Sumber Daya Amazon \(ARN\) untuk AWS DMS](#)
- [Menggunakan AWS DMS dengan AWS layanan lain](#)

Tampilan tingkat tinggi AWS DMS

Untuk melakukan migrasi database, AWS DMS sambungkan ke penyimpanan data sumber, baca data sumber, dan format data untuk konsumsi oleh penyimpanan data target. Kemudian memuat data ke tempat menyimpan data target. Sebagian besar pengolahan ini terjadi dalam memori, meskipun transaksi large mungkin memerlukan beberapa buffering pada disk. Transaksi cache dan mencatat file juga ditulis pada disk.

Pada tingkat tinggi, saat menggunakan AWS DMS Anda melakukan hal berikut:

- Temukan database di lingkungan jaringan Anda yang merupakan kandidat yang baik untuk migrasi.
- Secara otomatis mengkonversi skema database sumber Anda dan sebagian besar objek kode database ke format yang kompatibel dengan database target.
- Membuat server replikasi.
- Membuat titik akhir sumber dan target yang memiliki informasi koneksi tentang penyimpanan data Anda.
- Membuat satu atau beberapa tugas migrasi untuk melakukan migrasi data antara penyimpanan data sumber dan target.

Sebuah tugas dapat terdiri dari tiga tahap utama:

- Migrasi data yang ada (Full load)
- Aplikasi perubahan cache
- Replikasi yang sedang berlangsung (Ubah Pengambilan Data)

Selama migrasi beban penuh, di mana data yang ada dari sumber dipindahkan ke target, AWS DMS memuat data dari tabel di penyimpanan data sumber ke tabel di penyimpanan data target. Sementara beban penuh sedang berlangsung, setiap perubahan yang dibuat untuk tabel dimuat cache pada server replikasi; ini adalah perubahan cache. Penting untuk dicatat bahwa AWS DMS tidak menangkap perubahan untuk tabel tertentu sampai beban penuh untuk tabel tersebut dimulai. Dengan kata lain, poin ketika perubahan capture dimulai berbeda untuk setiap tabel individu.

Ketika beban penuh untuk tabel tertentu selesai, AWS DMS segera mulai menerapkan perubahan cache untuk tabel itu. Setelah tabel dimuat dan perubahan cache diterapkan, AWS DMS mulai mengumpulkan perubahan sebagai transaksi untuk fase replikasi yang sedang berlangsung. Jika transaksi memiliki tabel yang belum sepenuhnya dimuat, perubahan disimpan secara lokal pada instans replikasi. Setelah AWS DMS menerapkan semua perubahan yang di-cache, tabel konsisten secara transaksional. Pada titik ini, AWS DMS pindah ke fase replikasi yang sedang berlangsung, menerapkan perubahan sebagai transaksi.

Saat fase replikasi mulai berlangsung, backlog transaksi umumnya menyebabkan beberapa kelambatan antara basis data sumber dan target. Migrasi akhirnya mencapai kondisi stabil setelah bekerja melalui backlog transaksi ini. Pada titik ini, Anda dapat menutup aplikasi, mengizinkan setiap transaksi yang tersisa untuk diterapkan pada target, dan memunculkan aplikasi Anda, sekarang menunjuk pada basis data target.

AWS DMS menciptakan objek skema target yang diperlukan untuk melakukan migrasi data. Anda dapat AWS DMS menggunakan pendekatan minimalis dan hanya membuat objek yang diperlukan untuk memigrasikan data secara efisien. Dengan menggunakan pendekatan ini, AWS DMS membuat tabel, kunci utama, dan dalam beberapa kasus indeks unik, tetapi tidak akan membuat objek lain yang tidak diperlukan untuk memigrasikan data secara efisien dari sumber.

Atau, Anda dapat menggunakan Konversi Skema DMS dalam AWS DMS untuk secara otomatis mengonversi skema basis data sumber Anda dan sebagian besar objek kode database ke format yang kompatibel dengan database target. Konversi ini mencakup tabel, tampilan, prosedur tersimpan, fungsi, tipe data, sinonim, dan sebagainya. Objek apa pun yang Konversi Skema DMS tidak dapat dikonversi secara otomatis ditandai dengan jelas. Untuk menyelesaikan migrasi, Anda dapat mengonversi objek ini secara manual.

Komponen AWS DMS

Bagian ini menjelaskan komponen internal AWS DMS dan bagaimana mereka berfungsi bersama untuk menyelesaikan migrasi data Anda. Memahami komponen dasar AWS DMS dapat membantu Anda melakukan migrasi data lebih efisien dan memberikan wawasan yang lebih baik saat memecahkan masalah atau menyelidiki masalah.

AWS DMS Migrasi terdiri dari lima komponen: penemuan database untuk bermigrasi, konversi skema otomatis, contoh replikasi, titik akhir sumber dan target, dan tugas replikasi. Anda membuat AWS DMS migrasi dengan membuat instance replikasi, titik akhir, dan tugas yang diperlukan dalam file. Wilayah AWS

Penemuan basis data

DMS Fleet Advisor mengumpulkan data dari beberapa lingkungan database untuk memberikan wawasan tentang infrastruktur data Anda. DMS Fleet Advisor mengumpulkan data dari database lokal dan server analitik Anda dari satu atau beberapa lokasi pusat tanpa perlu menginstalnya di setiap komputer. Saat ini, DMS Fleet Advisor mendukung server database Microsoft SQL Server, MySQL, Oracle, dan PostgreSQL.

Berdasarkan data yang ditemukan dari jaringan Anda, DMS Fleet Advisor membangun inventaris yang dapat Anda tinjau untuk menentukan server dan objek database mana yang akan dipantau. Saat detail tentang server, database, dan skema ini dikumpulkan, Anda dapat menganalisis kelayakan migrasi basis data yang Anda inginkan.

Migrasi skema dan kode

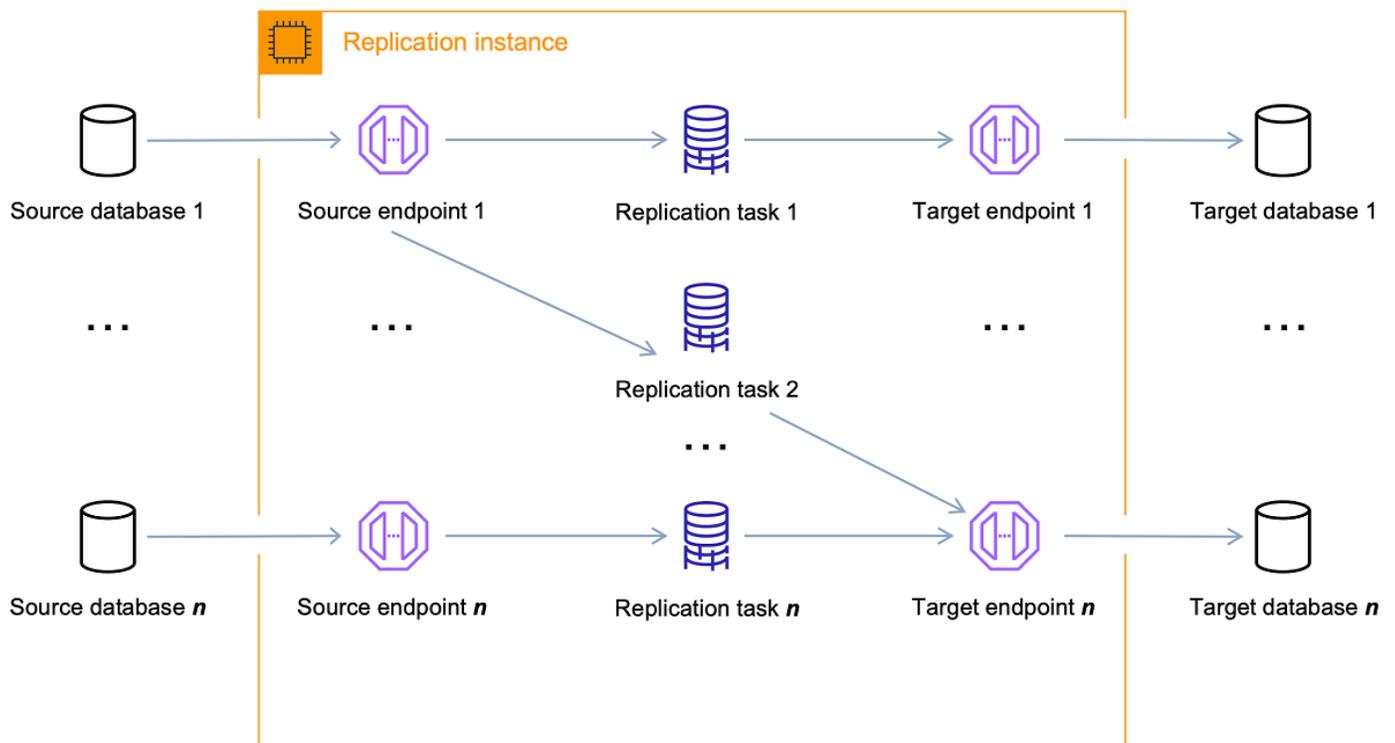
Konversi Skema DMS AWS DMS membuat migrasi database antara berbagai jenis database lebih dapat diprediksi. Anda dapat menggunakan Konversi Skema DMS untuk mengevaluasi kompleksitas migrasi Anda untuk penyedia data sumber Anda, dan kemudian menggunakannya untuk mengonversi skema database dan objek kode. Anda kemudian dapat menerapkan kode yang dikonversi ke database target Anda.

Pada tingkat tinggi, Konversi Skema DMS beroperasi dengan tiga komponen berikut: profil instance, penyedia data, dan proyek migrasi. Profil instance menentukan pengaturan jaringan dan keamanan. Penyedia data menyimpan kredensial koneksi database. Proyek migrasi berisi penyedia data, profil instance, dan aturan migrasi. AWS DMS menggunakan penyedia data dan profil instance untuk merancang proses yang mengubah skema database dan objek kode.

Instans replikasi

Pada tingkat tinggi, instans AWS DMS replikasi hanyalah instans Amazon Elastic Compute Cloud (Amazon EC2) terkelola yang menampung satu atau beberapa tugas replikasi.

Gambar berikut menunjukkan contoh replikasi yang menjalankan beberapa tugas replikasi terkait.



Sebuah contoh replikasi tunggal dapat meng-host satu atau beberapa tugas replikasi, tergantung pada karakteristik migrasi Anda dan kapasitas server replikasi. AWS DMS menyediakan berbagai contoh replikasi sehingga Anda dapat memilih konfigurasi optimal untuk kasus penggunaan Anda. Untuk informasi lebih lanjut tentang berbagai kelas instans replikasi, lihat [Memilih instans replikasi AWS DMS yang tepat untuk migrasi Anda](#).

AWS DMS membuat instance replikasi pada instans Amazon EC2. Beberapa kelas instans yang lebih kecil cukup untuk menguji layanan atau untuk migrasi kecil. Jika migrasi Anda melibatkan jumlah tabel large, atau jika Anda berniat menjalankan beberapa tugas replikasi secara bersamaan, Anda harus mempertimbangkan penggunaan salah satu instans yang lebih besar. Kami merekomendasikan pendekatan ini karena AWS DMS dapat menghabiskan sejumlah besar memori dan CPU.

Tergantung pada kelas instans Amazon EC2 pilihan Anda, instans replikasi Anda dilengkapi dengan 50 GB atau 100 GB penyimpanan data. Jumlah ini biasanya mencukupi untuk kebanyakan pelanggan. Namun, jika migrasi Anda melibatkan transaksi large atau volume perubahan data yang tinggi maka Anda mungkin ingin meningkatkan alokasi penyimpanan dasar. Change data capture (CDC) dapat menyebabkan data tulis pada disk, tergantung pada seberapa cepat target tulis perubahan. Karena file log juga ditulis ke disk, meningkatkan tingkat keparahan untuk logging juga akan menyebabkan konsumsi penyimpanan yang lebih tinggi.

AWS DMS dapat memberikan ketersediaan tinggi dan dukungan failover menggunakan penerapan Multi-AZ. Dalam penerapan Multi-AZ, AWS DMS secara otomatis menyediakan dan memelihara replika siaga instans replikasi di Availability Zone yang berbeda. Instans replikasi utama secara sinkron dilakukan replikasi untuk replika cadangan. Jika instans replikasi utama gagal atau menjadi tidak responsif, cadangan tersebut melanjutkan tugas yang sedang berlangsung dengan gangguan minimal. Karena instans replikasi utama terus-menerus mereplikasi keadaan pada cadangannya, deployment multi-AZ tidak dikenakan beberapa overhead performa.

Untuk informasi lebih rinci tentang contoh AWS DMS replikasi, lihat [Bekerja dengan instance AWS DMS replikasi](#).

Daripada membuat dan mengelola instance replikasi, Anda dapat membiarkan AWS DMS penyediaan replikasi Anda secara otomatis menggunakan AWS DMS Tanpa Server. Untuk informasi selengkapnya, lihat [Bekerja dengan Tanpa AWS DMS Server](#).

Titik Akhir

AWS DMS menggunakan titik akhir untuk mengakses sumber atau penyimpanan data target Anda. Informasi koneksi tertentu akan berbeda, tergantung pada tempat menyimpan data Anda, tetapi secara umum Anda menyediakan informasi berikut ketika membuat titik akhir:

- Tipe titik akhir - Sumber atau target.
- Jenis mesin — Jenis mesin database, seperti Oracle atau PostgreSQL.
- Nama server — Nama server atau alamat IP yang AWS DMS dapat menjangkau.
- Port - Nomor port yang digunakan untuk koneksi server basis data.
- Enkripsi – mode Lapisan Soket Aman (SSL), jika SSL digunakan untuk mengenkripsi koneksi.
- Kredensial—Nama pengguna dan kata sandi untuk suatu akun dengan hak mengakses yang diperlukan.

Saat Anda membuat titik akhir menggunakan AWS DMS konsol, konsol mengharuskan Anda menguji koneksi titik akhir. Tes harus berhasil sebelum menggunakan titik akhir dalam suatu AWS DMS tugas. Seperti informasi koneksi, kriteria uji spesifik berbeda untuk tipe mesin yang berbeda. Secara umum, AWS DMS memverifikasi bahwa basis data terdapat pada nama server dan port tertentu, dan bahwa kredensial yang disediakan dapat digunakan untuk menjalankan connect ke basis data dengan hak istimewa yang diperlukan untuk melakukan suatu migrasi. Jika tes koneksi berhasil, AWS DMS unduh dan simpan informasi skema untuk digunakan nanti selama konfigurasi tugas. Informasi skema mungkin termasuk misalnya definisi tabel, definisi kunci primer, dan definisi kunci yang unik.

Lebih dari satu replikasi tugas dapat menggunakan satu titik akhir. Misalnya, Anda mungkin memiliki dua aplikasi yang secara logika berbeda yang dijadikan host pada basis data sumber yang sama yang ingin Anda migrasikan secara terpisah. Dalam kasus ini, Anda membuat dua tugas replikasi, satu untuk setiap set tabel aplikasi. Anda dapat menggunakan AWS DMS titik akhir yang sama di kedua tugas.

Anda dapat menyesuaikan perilaku titik akhir dengan menggunakan pengaturan titik akhir. Pengaturan titik akhir dapat mengontrol berbagai perilaku seperti detail logging, ukuran file, dan parameter lainnya. Setiap jenis mesin penyimpanan data memiliki pengaturan titik akhir yang berbeda yang tersedia. Anda dapat menemukan pengaturan titik akhir tertentu untuk setiap penyimpanan data di bagian sumber atau target untuk penyimpanan data tersebut. Untuk daftar penyimpanan data sumber dan target yang didukung, lihat [Sumber untuk AWS DMS](#) dan [Target untuk AWS DMS](#).

Untuk informasi lebih rinci tentang AWS DMS titik akhir, lihat [Bekerja dengan Titik akhir DMS AWS](#).

Tugas replikasi

Anda menggunakan tugas AWS DMS replikasi untuk memindahkan sekumpulan data dari titik akhir sumber ke titik akhir target. Membuat satu tugas replikasi adalah langkah terakhir yang perlu Anda ambil sebelum mulai migrasi.

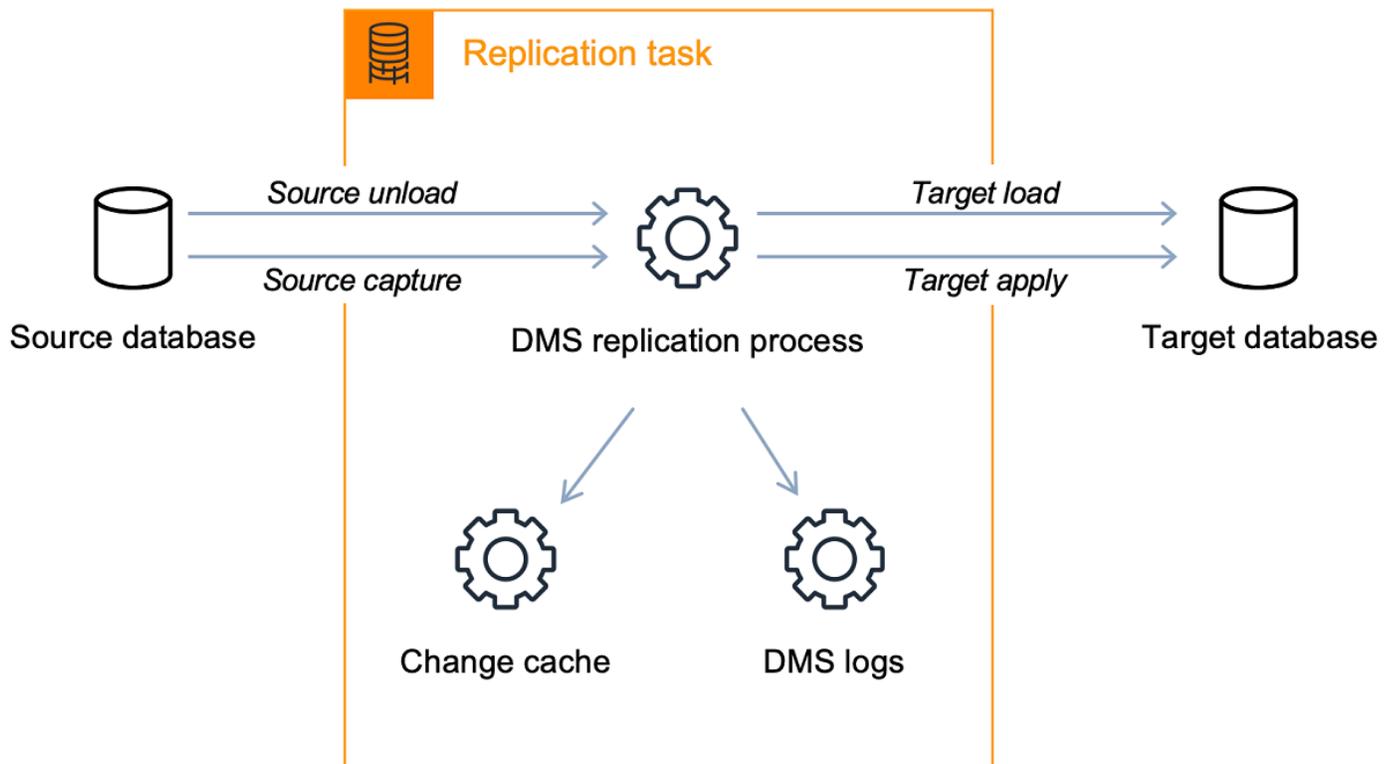
Ketika membuat tugas replikasi, Anda menentukan pengaturan tugas berikut:

- Instans replikasi - instans untuk menjadi host dan menjalankan tugas
- Titik akhir sumber
- Titik akhir target
- Opsi jenis migrasi, seperti yang tercantum berikut. Untuk penjelasan lengkap tentang opsi tipe migrasi, lihat [Membuat tugas](#).
 - Beban penuh (Migrasi data yang ada) – Jika Anda mampu melakukan penghentian pekerjaan yang cukup lama untuk menyalin data yang ada, memilih opsi ini merupakan pilihan yang tepat. Opsi ini hanya melakukan migrasi data dari basis data sumber ke basis data target Anda, membuat tabel-tabel bila diperlukan.
 - Beban penuh + CDC (Memigrasi data yang ada dan mereplikasi perubahan yang sedang berlangsung) - Opsi ini melakukan beban data penuh saat menangkap perubahan pada sumber. Setelah beban penuh selesai, perubahan yang ditangkap diterapkan pada target. Akhirnya, aplikasi perubahan mencapai keadaan stabil. Pada titik ini, Anda dapat menutup aplikasi Anda, membiarkan perubahan yang tersisa mengalir ke target, dan kemudian ulang kembali aplikasi Anda yang menunjuk pada target.
 - Hanya CDC (Mereplikasi perubahan data saja) - Dalam beberapa situasi, mungkin lebih efisien untuk menyalin data yang ada menggunakan metode selain AWS DMS. Misalnya, dalam migrasi homogen, menggunakan perangkat ekspor dan impor yang asli mungkin lebih efisien dalam memuat data dalam jumlah besar. Dalam situasi ini, Anda dapat menggunakan AWS DMS untuk mereplikasi perubahan yang dimulai saat Anda memulai pemuatan massal untuk membawa dan menjaga basis data sumber dan target Anda tetap sinkron.
- Pilihan mode persiapan tabel target, seperti yang tercantum berikut. Untuk penjelasan lengkap tentang mode tabel target, lihat [Membuat tugas](#).
 - Jangan lakukan apa-apa — AWS DMS mengasumsikan bahwa tabel target dibuat sebelumnya pada target.
 - Jatuhkan tabel pada target — AWS DMS turun dan buat ulang tabel target.

- Memotong – Jika Anda membuat tabel pada target, DMS AWS memotong tabel tersebut sebelum migrasi dimulai. Jika tidak ada tabel dan Anda memilih opsi ini, AWS DMS buat tabel yang hilang.
- Pilihan mode LOB, seperti yang tercantum berikut. Untuk penjelasan lengkap tentang mode LOB, lihat [Mengatur dukungan LOB untuk database sumber dalam suatu tugas AWS DMS](#).
 - Jangan sertakan kolom LOB – Kolom LOB tidak dimasukkan dalam proses migrasi.
 - Mode LOB penuh - Migrasikan LOB lengkap terlepas dari ukurannya. AWS DMS memigrasikan LOB sedikit demi sedikit dalam potongan yang dikontrol oleh parameter Ukuran LOB Maks. Mode ini lebih lambat daripada menggunakan mode LOB terbatas.
 - Mode LOB terbatas – Memotong LOB pada nilai yang ditetapkan oleh parameter Ukuran LOB maksimal. Mode ini lebih cepat daripada menggunakan mode LOB penuh.
- Pemetaan tabel – menunjukkan tabel-tabel bermigrasi dan bagaimana tabel tersebut bermigrasi. Untuk informasi selengkapnya, lihat [Menggunakan pemetaan tabel untuk menentukan pengaturan tugas](#).
- Transformasi data, seperti yang tercantum berikut. Untuk informasi lebih lanjut tentang transformasi data, lihat [Menentukan pemilihan tabel dan transformasi aturan menggunakan JSON](#).
 - Mengubah nama skema, tabel, dan kolom.
 - Mengubah nama tempat penyimpanan logikal pada basis data (untuk titik akhir target Oracle).
 - Menetapkan kunci primer dan indeks unik pada target.
- Validasi data
- CloudWatch Pencatatan Amazon

Anda menggunakan tugas untuk melakukan migrasi data dari titik akhir sumber ke titik akhir target, dan pemrosesan tugas dilakukan pada instans replikasi. Anda menentukan tabel dan skema mana yang bermigrasi dan setiap pemrosesan khusus, seperti persyaratan pencatatan, tabel data kontrol, dan penanganan kesalahan.

Secara konseptual, tugas AWS DMS replikasi melakukan dua fungsi yang berbeda seperti yang ditunjukkan pada diagram berikut.



Proses beban penuh cukup mudah dipahami. Data diekstraksi dari sumber dengan cara mengekstraksi dalam jumlah banyak dan dimuat langsung ke target. Anda dapat menentukan jumlah tabel yang akan diekstrak dan dimuat secara paralel di AWS DMS konsol di bawah Pengaturan Lanjutan.

Untuk informasi selengkapnya tentang AWS DMS tugas, lihat [Bekerja dengan AWS DMS tugas](#).

Replikasi yang sedang berlangsung, atau change data capture (CDC)

Anda juga dapat menggunakan AWS DMS tugas untuk menangkap perubahan yang sedang berlangsung pada penyimpanan data sumber saat Anda memigrasikan data ke target. Proses pengambilan perubahan yang AWS DMS digunakan saat mereplikasi perubahan yang sedang berlangsung dari titik akhir sumber mengumpulkan perubahan pada log database dengan menggunakan API asli mesin database.

Dalam proses CDC, tugas replikasi dirancang untuk pengaliran perubahan dari sumber ke target, menggunakan buffer dalam memori untuk menyimpan data dalam transit. Jika buffer dalam memori menjadi habis karena alasan apa pun, tugas replikasi akan menginformasikan perubahan yang tertunda ke Cache Perubahan pada disk. Ini bisa terjadi, misalnya, jika AWS DMS menangkap perubahan dari sumber lebih cepat daripada yang dapat diterapkan pada target. Dalam hal ini, Anda akan melihat Latensi target tugas melebihi Latensi sumber tugas.

Anda dapat memeriksa ini dengan menavigasi ke tugas Anda di AWS DMS konsol, dan membuka tab Pemantauan Tugas. LatencySourceGrafik CDC LatencyTarget dan CDC ditampilkan di bagian bawah halaman. Jika Anda memiliki tugas yang menunjukkan latensi target maka ada kemungkinan beberapa penyetelan pada titik akhir target yang diperlukan untuk meningkatkan nilai aplikasi.

Tugas replikasi juga menggunakan penyimpanan untuk pencatatan tugas seperti yang dibahas sebelumnya. Ruang disk yang dilengkapi oleh pra-dikonfigurasi dengan instans replikasi Anda biasanya cukup untuk pencatatan dan perubahan yang diinformasikan. Jika Anda memerlukan ruang disk tambahan, misalnya, ketika menggunakan debugging detail untuk menyelidiki masalah migrasi, Anda dapat memodifikasi instans replikasi untuk mengalokasikan lebih banyak ruang.

Sumber untuk AWS DMS

Anda dapat menggunakan penyimpanan data sumber yang berbeda dalam AWS DMS fitur yang berbeda. Bagian berikut berisi daftar penyimpanan data sumber yang didukung untuk setiap AWS DMS fitur.

Topik

- [Titik akhir sumber untuk migrasi data](#)
- [Database sumber untuk DMS Fleet Advisor](#)
- [Penyedia data sumber untuk Konversi Skema DMS](#)
- [Penyedia data sumber untuk migrasi data homogen DMS](#)

Titik akhir sumber untuk migrasi data

Anda dapat menggunakan penyimpanan data berikut sebagai titik akhir sumber untuk migrasi data dengan menggunakan AWS DMS.

Basis data on premise dan Instans EC2

- Oracle versi 10.2 dan lebih tinggi (untuk versi 10.x), 11g dan hingga 12.2, 18c, dan 19c untuk edisi Enterprise, Standard, Standard One, dan Standard Two
- Microsoft SQL Server versi 2005, 2008, 2008R2, 2012, 2014, 2016, 2017, 2019, dan 2022.
 - Edisi Enterprise, Standard, Workgroup, Developer, dan Web mendukung replikasi beban penuh.

- Edisi Enterprise, Standard (versi 2016 dan lebih tinggi), dan Developer mendukung replikasi CDC (berkelanjutan) selain beban penuh.
- Edisi Express tidak didukung.
- MySQL versi 5.5, 5.6, 5.7, dan 8.0

 Note

Support untuk MySQL 8.0 sebagai sumber tersedia AWS DMS dalam versi 3.4.0 dan lebih tinggi, kecuali ketika muatan transaksi dikompresi. Support untuk Google Cloud untuk MySQL 8.0 sebagai sumber tersedia AWS DMS dalam versi 3.4.6 dan lebih tinggi.

- MariaDB (didukung sebagai sumber data yang kompatibel dengan MySQL) versi 10.0 (hanya versi 10.0.24 dan lebih tinggi), 10.2, 10.3, 10.4, 10.5, dan 10.6.

 Note

Dukungan untuk MariaDB sebagai sumber tersedia di AWS DMS semua versi di mana MySQL didukung.

- PostgreSQL versi 9.4 dan lebih tinggi (untuk versi 9.x), 10.x, 11.x, 12.x, 13.x 14.x, dan 15.x.

 Note

AWS DMS hanya mendukung PostgreSQL versi 15.x di versi 3.5.1 dan lebih tinggi.

- MongoDB versi 3.x, 4.0, 4.2, 4.4, 5.0, dan 6.0

 Note

AWS DMS versi 3.5.0 dan yang lebih tinggi tidak mendukung versi MongoDB sebelum 3.6.

- SAP Adaptive Server Enterprise (ASE) versi 12.5, 15, 15.5, 15.7, 16, dan lebih tinggi
- IBM Db2 untuk versi Linux, UNIX dan Windows (Db2 LUW):
 - Versi 9.7, semua paket perbaikan
 - Versi 10.1, semua paket perbaikan
 - Versi 10.5, semua paket perbaikan kecuali untuk Fix Pack 5
 - Versi 11.1, semua paket perbaikan

- Versi 11.5, Mods (0-8) dengan hanya Fix Pack Zero
- IBM Db2 untuk z/OS versi 12

Layanan database terkelola pihak ketiga:

- Database Microsoft Azure SQL
- Microsoft Azure PostgreSQL Server Fleksibel versi 11.2, 12.15, 13.11, 14.8, dan 15.3.
- Microsoft Azure MySQL Server Fleksibel versi 5.7 dan 8.
- Google Cloud untuk MySQL versi 5.6, 5.7, dan 8.0.
- Google Cloud untuk PostgreSQL versi 9.6, 10, 11, 12, 13, 14, dan 15.
- OCI MySQL Heatwave versi 8.0.34.

Instans basis data Amazon RDS, dan Amazon Simple Storage Service (Amazon S3)

- Oracle versi 11g (versi 11.2.0.4 dan lebih tinggi) dan hingga 12.2, 18c, dan 19c untuk edisi Enterprise, Standard, Standard One, dan Standard Two
- Microsoft SQL Server versi 2012, 2014, 2016, 2017, 2019, dan 2022 untuk edisi Enterprise, Standard, Workgroup, dan Developer

 Note

AWS DMS tidak mendukung SQL Server Express. Edisi Web hanya didukung untuk replikasi full-load saja.

- MySQL versi 5.5, 5.6, 5.7, dan 8.0

 Note

Support untuk MySQL 8.0 sebagai sumber tersedia AWS DMS dalam versi 3.4.0 dan lebih tinggi, kecuali ketika muatan transaksi dikompresi.

- MariaDB (didukung sebagai sumber data yang kompatibel dengan MySQL) versi 10.0.24 hingga 10.0.28, 10.2, 10.3, 10.4, 10.5, dan 10.6.

Note

Dukungan untuk MariaDB sebagai sumber tersedia di AWS DMS semua versi di mana MySQL didukung.

- PostgreSQL versi 10.x, 11.x, 12.x, 13.x, 14.x, dan 15.x.

Note

AWS DMS hanya mendukung PostgreSQL 15.x di versi 3.5.1 dan lebih tinggi.

- Amazon Aurora dengan kompatibilitas MySQL (didukung sebagai sumber data yang kompatibel dengan MySQL)
- Amazon Aurora dengan kompatibilitas PostgreSQL (didukung sebagai sumber data yang kompatibel dengan PostgreSQL)
- Amazon S3
- Amazon DocumentDB (dengan kompatibilitas MongoDB) versi 3.6, 4.0, dan 5.0.
- Amazon RDS untuk IBM Db2 LUW.

Untuk informasi tentang bekerja dengan sumber tertentu, lihat [Bekerja dengan AWS DMS titik akhir](#).

Untuk informasi tentang titik akhir target yang didukung, lihat [Target titik akhir untuk migrasi data](#).

Database sumber untuk DMS Fleet Advisor

DMS Fleet Advisor mendukung database sumber berikut.

- Microsoft SQL Server versi 2012 dan hingga 2019
- MySQL versi 5.6 dan hingga 8
- Oracle versi 11g Rilis 2 dan hingga 12c, 19c, dan 21c
- PostgreSQL versi 9.6 dan hingga 13

Untuk informasi tentang bekerja dengan sumber tertentu, lihat [Membuat pengguna database untuk AWS DMS Fleet Advisor](#).

Untuk daftar database yang digunakan DMS Fleet Advisor untuk menghasilkan rekomendasi target, lihat [Target untuk DMS Fleet Advisor](#)

Penyedia data sumber untuk Konversi Skema DMS

Konversi Skema DMS mendukung penyedia data berikut sebagai sumber untuk proyek migrasi Anda.

- Microsoft SQL Server versi 2008 R2, 2012, 2014, 2016, 2017, dan 2019
- Oracle versi 10.2 dan lebih tinggi, 11g dan hingga 12.2, 18c, dan 19c, dan Oracle Data Warehouse
- PostgreSQL versi 9.2 dan lebih tinggi
- MySQL versi 5.5 dan lebih tinggi

Penyedia data sumber Anda dapat berupa engine yang dikelola sendiri yang berjalan di lokasi atau di instans Amazon Elastic Compute Cloud (Amazon EC2).

Untuk informasi tentang bekerja dengan sumber tertentu, lihat [Membuat penyedia data sumber dalam Konversi Skema DMS](#).

Untuk informasi tentang basis data target yang didukung, lihat [Penyedia data target untuk Konversi Skema DMS](#).

The AWS Schema Conversion Tool (AWS SCT) mendukung lebih banyak basis data sumber dan target daripada Konversi Skema DMS. Untuk informasi tentang database yang AWS SCT mendukung, lihat [Apa itu. AWS Schema Conversion Tool](#)

Penyedia data sumber untuk migrasi data homogen DMS

Anda dapat menggunakan penyedia data berikut sebagai sumber untuk migrasi data yang homogen.

- MySQL versi 5.7 dan lebih tinggi
- MariaDB versi 10.2 dan lebih tinggi
- PostgreSQL versi 10.4 hingga 14.x.
- MongoDB versi 4.x, 5.x, 6.0
- Amazon DocumentDB versi 3.6, 4.0, 5.0

Penyedia data sumber Anda dapat berupa mesin yang dikelola sendiri yang berjalan di lokasi atau di instans Amazon EC2. Selain itu, Anda dapat menggunakan instans Amazon RDS DB sebagai penyedia data sumber.

Untuk informasi tentang bekerja dengan sumber tertentu, lihat [Membuat penyedia data sumber untuk migrasi data homogen di AWS DMS](#).

Untuk informasi tentang basis data target yang didukung, lihat [Penyedia data target untuk migrasi data homogen DMS](#).

Target untuk AWS DMS

Anda dapat menggunakan penyimpanan data target yang berbeda dalam AWS DMS fitur yang berbeda. Bagian berikut berisi daftar penyimpanan data target yang didukung untuk setiap AWS DMS fitur.

Topik

- [Target titik akhir untuk migrasi data](#)
- [Basis data target untuk DMS Fleet Advisor](#)
- [Penyedia data target untuk Konversi Skema DMS](#)
- [Penyedia data target untuk migrasi data homogen DMS](#)

Target titik akhir untuk migrasi data

Anda dapat menggunakan tempat menyimpan data berikut sebagai titik akhir target untuk migrasi data dengan menggunakan AWS DMS.

Basis data on premise dan Instans Amazon EC2

- Oracle versi 10g, 11g, 12c, 18c, dan 19c untuk edisi Enterprise, Standard, Standard One, dan Standard Two
- Microsoft SQL Server versi 2005, 2008, 2008R2, 2012, 2014, 2016, 2017, 2019, dan 2022 untuk edisi Enterprise, Standard, Workgroup, dan Developer

Note

AWS DMS tidak mendukung edisi SQL Server Web dan Express.

- MySQL versi 5.5, 5.6, 5.7, dan 8.0
- MariaDB (didukung sebagai target data yang kompatibel dengan MySQL) versi 10.0.24 hingga 10.0.28, 10.2, 10.3, 10.4, 10.5, dan 10.6.

Note

Support untuk MariaDB sebagai target tersedia di AWS DMS semua versi di mana MySQL didukung.

- PostgreSQL versi 9.4 dan lebih tinggi (untuk versi 9.x), 10.x, 11.x, 12.x, 13.x, 14.x, dan 15.x.

Note

AWS DMS hanya mendukung PostgreSQL 15.x di versi 3.5.1 dan lebih tinggi.

- SAP Adaptive Server Enterprise (ASE) versi 15, 15.5, 15.7, 16, dan lebih tinggi
- Redis versi 6.x

Basis data instans Amazon RDS, Amazon Redshift, Amazon Redshift Tanpa Server, Amazon DynamoDB, Amazon S3, Layanan Amazon, Amazon untuk Redis, Amazon Kinesis Data Streams, Amazon DocumentDB, OpenSearch Amazon Neptunus, ElastiCache dan Apache Kafka

- Oracle versi 11g (versi 11.2.0.3.v1 dan lebih tinggi), 12c, 18c, dan 19c untuk edisi Enterprise, Standard, Standard One, dan Standard Two
- Microsoft SQL Server versi 2012, 2014, 2016, 2017, 2019, dan 2022 untuk edisi Enterprise, Standard, Workgroup, dan Developer

Note

AWS DMS tidak mendukung edisi SQL Server Web dan Express.

- MySQL versi 5.5, 5.6, 5.7, dan 8.0
- MariaDB (didukung sebagai target data yang kompatibel dengan MySQL) versi 10.0.24 hingga 10.0.28, 10.2, 10.3, 10.4, 10.5, dan 10.6.

Note

Support untuk MariaDB sebagai target tersedia di AWS DMS semua versi di mana MySQL didukung.

- PostgreSQL versi 10.x, 11.x, 12.x, 13.x, 14.x, dan 15.x

Note

AWS DMS hanya mendukung PostgreSQL 15.x di versi 3.5.1 dan lebih tinggi.

- IBM Db2 LUW versi 11.1 dan 11.5
- Edisi yang Kompatibel dengan Amazon Aurora MySQL
- Edisi yang Kompatibel dengan Amazon Aurora PostgreSQL
- Amazon Aurora Tanpa Server v2
- Amazon Redshift
- Amazon Redshift Tanpa Server
- Amazon S3
- Amazon DynamoDB
- OpenSearch Layanan Amazon
- Amazon ElastiCache untuk Redis
- Amazon Kinesis Data Streams
- Gunakan Amazon DocumentDB (dengan kompatibilitas MongoDB)
- Amazon Neptune
- Apache Kafka - [Amazon Managed Streaming for Apache Kafka \(Amazon MSK\)](#) dan [Apache Kafka yang dikelola sendiri](#)
- Babelfish (versi 3.2.0 dan lebih tinggi) untuk Aurora PostgreSQL (versi 15.3/14.8 dan lebih tinggi)

Untuk informasi tentang bekerja dengan target tertentu, lihat [Bekerja dengan AWS DMS titik akhir](#).

Untuk informasi tentang titik akhir sumber yang didukung, lihat [Titik akhir sumber untuk migrasi data](#).

Basis data target untuk DMS Fleet Advisor

DMS Fleet Advisor menghasilkan rekomendasi target menggunakan versi terbaru dari database target berikut.

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

- Amazon RDS for PostgreSQL
- Amazon RDS for SQL Server

Untuk informasi tentang rekomendasi target di DMS Fleet Advisor, lihat [Menggunakan fitur Rekomendasi Target Penasihat AWS DMS Armada](#)

Untuk informasi tentang basis data sumber yang didukung, lihat [Database sumber untuk DMS Fleet Advisor](#).

Penyedia data target untuk Konversi Skema DMS

Konversi Skema DMS mendukung penyedia data berikut sebagai target untuk proyek migrasi Anda.

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

Untuk informasi tentang bekerja dengan target tertentu, lihat [Membuat penyedia data target dalam Konversi Skema DMS](#).

Untuk informasi tentang basis data sumber yang didukung, lihat [Penyedia data sumber untuk Konversi Skema DMS](#).

Penyedia data target untuk migrasi data homogen DMS

Anda dapat menggunakan penyedia data berikut sebagai target untuk migrasi data homogen.

- Amazon Aurora MySQL versi 5.7 dan lebih tinggi
- Amazon Aurora PostgreSQL versi 10.4 hingga 14.x
- Amazon Aurora Tanpa Server v2
- Amazon RDS for MySQL versi 5.7 dan lebih tinggi
- Amazon RDS untuk MariaDB versi 10.2 dan lebih tinggi
- Amazon RDS untuk PostgreSQL versi 10.4 hingga 14.x
- Amazon DocumentDB versi 4.0, 5.0 dan DocumentDB cluster elastis

Untuk informasi tentang bekerja dengan target tertentu, lihat [Membuat penyedia data target untuk migrasi data homogen di AWS DMS](#).

Untuk informasi tentang basis data sumber yang didukung, lihat [Penyedia data sumber untuk migrasi data homogen DMS](#).

Membangun Nama Sumber Daya Amazon (ARN) untuk AWS DMS

Jika Anda menggunakan AWS DMS API AWS CLI atau untuk mengotomatiskan migrasi database, maka Anda bekerja dengan Amazon Resource Name (ARN). Setiap sumber daya yang dibuat di Amazon Web Services diidentifikasi oleh ARN, yang merupakan pengidentifikasi unik. Jika Anda menggunakan AWS DMS API AWS CLI atau untuk menyiapkan migrasi database, Anda menyediakan ARN sumber daya yang ingin Anda gunakan.

ARN untuk AWS DMS sumber daya menggunakan sintaks berikut:

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

Dalam sintaks ini, hal berikut ini berlaku:

- *region* adalah ID Wilayah AWS tempat AWS DMS sumber daya dibuat, seperti us-west-2.

Tabel berikut menunjukkan Wilayah AWS nama dan nilai yang harus Anda gunakan saat membangun ARN.

Wilayah	Nama
Wilayah Asia Pacific (Tokyo)	ap-northeast-1
Wilayah Asia Pacific (Seoul)	ap-northeast-2
Wilayah Asia Pacific (Mumbai)	ap-south-1
Wilayah Asia Pacific (Singapore)	ap-southeast-1
Wilayah Asia Pacific (Sydney)	ap-southeast-2
Wilayah Canada (Central)	ca-central-1
Wilayah China (Beijing)	cn-north-1

Wilayah	Nama
Wilayah China (Ningxia)	cn-northwest-1
Wilayah Eropa (Stockholm)	eu-north-1
Wilayah Eropa (Milan)	eu-south-1
Wilayah EU (Frankfurt)	eu-central-1
Wilayah Eropa (Irlandia)	eu-west-1
Wilayah EU (London)	eu-west-2
Wilayah EU (Paris)	eu-west-3
Wilayah South America (Sao Paulo)	sa-east-1
Wilayah US East (N. Virginia)	us-east-1
Wilayah US East (Ohio)	us-east-2
Wilayah US West (N. California)	us-west-1
Wilayah US West (Oregon)	us-west-2

- *account number* adalah nomor akun Anda dengan tanda hubung dihilangkan. Untuk menemukan nomor akun Anda, masuk ke AWS akun Anda di <http://aws.amazon.com>, pilih Akun Saya/Konsol, lalu pilih Akun Saya.
- *resourcetype* adalah jenis sumber AWS DMS daya.

Tabel berikut menunjukkan jenis sumber daya yang akan digunakan saat membangun ARN untuk sumber daya tertentu AWS DMS .

AWS DMS jenis sumber daya	Format ARN
Instans replikasi	arn:aws:dms: <i>region</i> : <i>account</i> :rep: <i>resourcename</i>
Titik akhir	arn:aws:dms: <i>region</i> : <i>account</i> :endpoint: <i>resourcename</i>

AWS DMS jenis sumber daya	Format ARN
Tugas replikasi	<code>arn:aws:dms: <i>region</i>:<i>account</i>:task:<i>resourcename</i></code>
Grup Subnet	<code>arn:aws:dms: <i>region</i>:<i>account</i>:subgrp:<i>resourcename</i></code>

- *resourcename* adalah nama sumber daya yang ditetapkan untuk AWS DMS sumber daya. Ini merupakan rangkaian apa saja yang dihasilkan.

Tabel berikut menunjukkan contoh ARN untuk AWS DMS sumber daya. Di sini, kita asumsikan akun AWS yaitu 123456789012, yang dibuat di Wilayah US East (N. Virginia), dan memiliki nama sumber daya.

Tipe sumber daya	Sampel ARN
Instans replikasi	<code>arn:aws:dms:us-east-1:123456789012:rep:QLXQZ64MH7CXF4QCQMGRVYVXAI</code>
Titik akhir	<code>arn:aws:dms:us-east-1:123456789012:endpoint:D3HMZ2IGUCGFF3NTAXUXGF6S5A</code>
Tugas replikasi	<code>arn:aws:dms:us-east-1:123456789012:task:2PVREMWNPJYJCVU2IBPTOYTIV4</code>
Grup Subnet	<code>arn:aws:dms:us-east-1:123456789012:subgrp:test-tag-grp</code>

Menggunakan AWS DMS dengan AWS layanan lain

Anda dapat menggunakan AWS DMS dengan beberapa AWS layanan lain:

- Anda dapat menggunakan Instans Amazon EC2 atau Amazon RDS instans DB sebagai target untuk migrasi data.
- Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skema sumber dan kode SQL Anda menjadi skema target dan kode SQL yang setara.

- Anda dapat menggunakan Amazon S3 sebagai tempat penyimpanan untuk data Anda, atau Anda dapat menggunakannya sebagai langkah tingkat menengah pada saat migrasi sejumlah data large.
- Anda dapat menggunakan AWS CloudFormation untuk mengatur AWS sumber daya Anda untuk manajemen infrastruktur atau penyebaran. Misalnya, Anda dapat menyediakan AWS DMS sumber daya seperti contoh replikasi, tugas, sertifikat, dan titik akhir. Anda membuat template yang menjelaskan semua AWS sumber daya yang Anda inginkan, dan menyediakan serta AWS CloudFormation mengonfigurasi sumber daya tersebut untuk Anda.

AWS DMS dukungan untuk AWS CloudFormation

Anda dapat menyediakan AWS DMS sumber daya menggunakan AWS CloudFormation. AWS CloudFormation adalah layanan yang membantu Anda memodelkan dan menyiapkan AWS sumber daya Anda untuk manajemen atau penyebaran infrastruktur. Misalnya, Anda dapat menyediakan AWS DMS sumber daya seperti contoh replikasi, tugas, sertifikat, dan titik akhir. Anda membuat template yang menjelaskan semua AWS sumber daya yang Anda inginkan dan AWS CloudFormation ketentuan dan mengonfigurasi sumber daya tersebut untuk Anda.

Sebagai developer atau administrator sistem, Anda dapat membuat dan mengelola koleksi sumber daya ini yang kemudian dapat digunakan untuk tugas migrasi berulang atau men-deploy sumber daya ke organisasi Anda. Untuk informasi selengkapnya AWS CloudFormation, lihat [AWS CloudFormation konsep](#) di Panduan AWS CloudFormation Pengguna.

AWS DMS mendukung pembuatan AWS DMS sumber daya berikut menggunakan AWS CloudFormation:

- [AWS::DMS::Certificate](#)
- [AWS::DMS::Endpoint](#)
- [AWS::DMS::EventSubscription](#)
- [AWS::DMS::ReplicationInstance](#)
- [AWS::DMS::ReplicationSubnetKelompok](#)
- [AWS::DMS::ReplicationTask](#)

Memulai dengan AWS Database Migration Service

Dalam tutorial berikut, Anda dapat mengetahui bagaimana melakukan migrasi basis data dengan AWS Database Migration Service (AWS DMS).

Untuk melakukan migrasi basis data, lakukan langkah-langkah berikut:

1. Siapkan akun AWS Anda dengan mengikuti langkah-langkah di [Pengaturan untuk AWS Database Migration Service](#).
2. Membuat basis data sampel Anda dan klien Amazon EC2 untuk mengisi basis data sumber dan tes replikasi. Buat Virtual Private Cloud (VPC) berdasarkan layanan Amazon Virtual Private Cloud (Amazon VPC) untuk berisi sumber daya tutorial Anda. Untuk membuat sumber daya ini, ikuti langkah-langkah dalam [Prasyarat untuk AWS Database Migration Service](#).
3. Mengisi basis data sumber Anda menggunakan [contoh penulisan pembuatan basis data](#).
4. Gunakan Konversi Skema DMS atau AWS Schema Conversion Tool (AWS SCT) untuk mengubah skema dari basis data target ke basis data target ke basis data target. Untuk menggunakan Konversi Skema DMS, ikuti langkah-langkah di [Memulai dengan Konversi Skema DMS](#). Untuk mengubah skema dengan AWS SCT, ikuti langkah-langkah dalam [Migrasi skema](#).
5. Membuat instans replikasi untuk melakukan semua proses untuk migrasi. Untuk melakukan ini dan tugas-tugas berikut, ambil langkah-langkah dalam [Replikasi](#).
6. Tentukan titik akhir basis data sumber dan target. Untuk informasi tentang membuat titik akhir, lihat [Membuat titik akhir sumber dan target](#).
7. Membuat tugas untuk menentukan tabel dan proses replikasi apakah yang ingin Anda gunakan, dan mulai replikasi. Untuk informasi tentang membuat tugas migrasi database, lihat [Membuat tugas](#).
8. Memeriksa bahwa replikasi bekerja dengan menjalankan kueri pada target basis data.

Pengaturan untuk AWS Database Migration Service

Daftar Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar Akun AWS

1. Buka <https://portal.aws.amazon.com/billing/signup>.

2. Ikuti petunjuk secara online.

Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

Saat Anda mendaftar Akun AWS, Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, [tetapkan akses administratif ke pengguna administratif](#), dan hanya gunakan pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

AWS akan mengirimkan email konfirmasi kepada Anda setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun saat ini dan mengelola akun dengan mengunjungi <https://aws.amazon.com/> dan memilih Akun Saya.

Membuat pengguna administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Mengamankan Pengguna root akun AWS Anda

1. Masuk ke [AWS Management Console](#) sebagai pemilik akun dengan memilih Pengguna root dan memasukkan alamat email Akun AWS Anda. Pada halaman berikutnya, masukkan kata sandi Anda.

Untuk bantuan masuk menggunakan pengguna root, lihat [Masuk sebagai pengguna root](#) dalam Panduan Pengguna AWS Sign-In.

2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

Untuk petunjuk, lihat [Mengaktifkan perangkat MFA virtual untuk pengguna root Akun AWS Anda \(konsol\)](#) dalam Panduan Pengguna IAM.

Membuat pengguna administratif

1. Aktifkan Pusat Identitas IAM.

Untuk petunjuk, lihat [Mengaktifkan AWS IAM Identity Center](#) di Panduan AWS IAM Identity Center Pengguna.

2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna administratif.

Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat [Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM](#) di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email Anda saat Anda membuat pengguna Pusat Identitas IAM.

Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat [Masuk ke portal akses AWS](#) dalam Panduan Pengguna AWS Sign-In.

Prasyarat untuk AWS Database Migration Service

Dalam bagian ini, Anda dapat mempelajari tugas-tugas prasyarat untuk AWS DMS, seperti mengatur basis data sumber dan target. Sebagai bagian dari tugas-tugas ini, Anda juga menyiapkan virtual private cloud (VPC) berdasarkan layanan Amazon VPC untuk mengisi sumber daya Anda. Selain itu, Anda mengatur instans Amazon EC2 yang Anda gunakan untuk mengisi basis data sumber Anda dan memeriksa replikasi pada basis data target Anda.

Note

Mengisi basis data sumber memakan waktu hingga 45 menit.

Untuk tutorial ini, Anda membuat database MariaDB sebagai sumber Anda, dan database PostgreSQL sebagai target Anda. Skenario ini menggunakan mesin basis data biaya murah yang umum digunakan untuk menunjukkan replikasi. Menggunakan mesin basis data yang berbeda menunjukkan fitur AWS DMS untuk migrasi data antara platform heterogen.

Sumber daya dalam tutorial ini menggunakan Wilayah US West (Oregon). Jika Anda ingin menggunakan Wilayah AWS yang berbeda, sebaliknya tentukan Wilayah yang Anda pilih di mana pun US West (Oregon) muncul.

Note

Demi kesederhanaan, basis data yang Anda buat untuk tutorial ini tidak menggunakan enkripsi atau fitur keamanan lanjutan lainnya. Anda harus menggunakan fitur keamanan untuk menjaga basis data produksi Anda tetap aman. Untuk informasi selengkapnya, lihat [Keamanan dalam Amazon RDS](#).

Untuk langkah-langkah prasyarat, lihat topik berikut.

Topik

- [Buat VPC](#)
- [Buat grup parameter Amazon RDS](#)
- [Membuat basis data Amazon RDS sumber Anda](#)
- [Membuat basis data Amazon RDS](#)
- [Membuat klien Amazon EC2](#)
- [Mengisi basis data sumber Anda](#)

Buat VPC

Dalam bagian ini, Anda membuat VPC untuk mengisi perangkat sumber daya AWS Anda. Menggunakan VPC adalah praktik terbaik saat menggunakan sumber daya AWS, sehingga basis data Anda, Instans Amazon EC2, grup keamanan, dan sebagainya, secara logika tertata dan aman.

Menggunakan VPC untuk sumber daya tutorial Anda juga memastikan bahwa Anda menghapus semua sumber daya yang Anda gunakan setelah selesai dengan tutorial. Anda harus menghapus semua sumber daya yang berisi VPC sebelum Anda dapat menghapus VPC.

Buat VPC untuk digunakan dengan AWS DMS

1. Masuk ke AWS Management Console dan buka konsol Amazon VPC di <https://console.aws.amazon.com/vpc/>.
2. Pada panel navigasi, pilih Dasbor VPC, lalu pilih Buat VPC.
3. Pada halaman Buat VPC, masukkan opsi berikut:
 - Sumber daya untuk membuat: VPC dan lainnya

- Generasi otomatis tag nama: Pilih Buat otomatis, dan masukkan **DMSVPC**.
- Blok IPv4: **10.0.1.0/24**
- Blok IPv6 CIDR: Tidak ada blok CIDR IPv6
- Sewa: Default
- Jumlah zona ketersediaan: 2
- Jumlah subnet publik: 2
- Jumlah subnet pribadi: 2
- Gateway NAT (\$): Tidak ada
- Titik akhir VPC: Tidak ada

Memilih Membuat VPC.

4. Pada panel navigasi, memilih VPC Anda. Catat VPC ID untuk DMSVPC.
5. Pada panel navigasi, memilih Grup Keamanan.
6. Memilih grup yang bernama default yang memiliki ID VPC yang cocok dengan ID yang Anda catat untuk DMSVPC.
7. Memilih tab Aturan Masuk, lalu memilih Edit aturan masuk.
8. Memilih Menambahkan aturan. Tambahkan aturan tipe MySQL/Aurora dan pilih di mana-mana-IPv4 untuk Sumber.
9. Memilih Menambahkan aturan sekali lagi. Tambahkan aturan tipe PostgreSQL dan pilih Anywhere-IPv4 untuk Source.
10. Memilih Simpan aturan.

Buat grup parameter Amazon RDS

Untuk menentukan pengaturan basis data sumber dan target Anda untuk AWS DMS, gunakan grup parameter Amazon RDS. Untuk mengizinkan replikasi awal dan berkelanjutan antara setiap basis data Anda, pastikan untuk mengkonfigurasi berikut:

- Mencatat biner basis data sumber Anda, sehingga AWS DMS dapat menentukan pembaruan tambahan apa yang dibutuhkan untuk melakukan replikasi.
- Peran replikasi basis data target Anda, sehingga AWS DMS mengabaikan batasan kunci asing selama transfer data awal. Dengan pengaturan ini, AWS DMS dapat melakukan migrasi data yang rusak.

Untuk membuat grup parameter untuk digunakan dengan AWS DMS

1. Buka konsol Amazon RDS di <https://console.aws.amazon.com/rds/>.
2. Pada panel navigasi, memilih grup parameter.
3. Pada halaman grup parameter, memilih Membuat grup parameter.
4. Pada halaman Membuat grup parameter, memasukkan beberapa pengaturan berikut:
 - Keluarga kelompok parameter: mariadb10.6
 - Nama kelompok: **dms-mariadb-parameters**
 - Deskripsi: **Group for specifying binary log settings for replication**

Memilih Membuat.

5. Pada halaman grup Parameter, pilih dms-mariadb-parameters, dan pada dms-mariadb-parametershalaman, pilih Edit.
6. Tetapkan parameter berikut pada nilai-nilai berikut:
 - binlog_checksum: TIDAK ADA
 - binlog_format: BARIS

Memilih Simpan perubahan.

7. Pada halaman grup parameter, memilih Membuat grup parameter lagi.
8. Pada halaman Membuat grup parameter, memasukkan beberapa pengaturan berikut:
 - Keluarga kelompok parameter: postgres13
 - Nama kelompok: **dms-postgresql-parameters**
 - Deskripsi: **Group for specifying role setting for replication**

Memilih Membuat.

9. Pada halaman grup Parameter, pilih dms-postgresql-parameters.
10. Pada dms-postgresql-parametershalaman, pilih Edit, dan atur parameter session_replication_role ke replika. Perhatikan bahwa parameter session_replication_role tidak ada di halaman pertama parameter. Gunakan kontrol pagination atau kolom pencarian untuk menemukan parameter.
11. Memilih Simpan perubahan.

Membuat basis data Amazon RDS sumber Anda

Gunakan prosedur berikut untuk membuat basis data Amazon RDS sumber Anda.

Untuk membuat sumber Anda Amazon RDS untuk database MariaDB

1. Buka konsol Amazon RDS di <https://console.aws.amazon.com/rds/>.
2. Pada Dasbor, memilih Membuat basis data dalam bagian Basis data. Jangan memilih Buat Database di Opsi penyebaran Multi-AZ Amazon RDS baru untuk bagian MySQL dan PostgreSQL di bagian atas halaman.
3. Pada halaman Membuat basis data, menetapkan pilihan berikut:
 - Pilih metode pembuatan database: Pilih Buat Standar.
 - Opsi mesin: Untuk tipe Mesin, pilih MariaDB. Untuk Versi, biarkan MariaDB 10.6.14 dipilih.
 - Templat: Memilih Dev/Test.
 - Pengaturan:
 - Pengidentifikasi instans DB: Memasukkan **dms-mariadb**.
 - Di bagian Pengaturan kredensial, masukkan yang berikut ini:
 - Nama pengguna utama : Biarkan sebagai **admin**.
 - Biarkan Manage master credentials di AWS Secrets Manager tidak dicentang.
 - Buat kata sandi secara otomatis: Biarkan tanpa dipilih.
 - Kata sandi utama: Memasukkan **changeit**.
 - Mengonfirmasi kata sandi: memasukkan **changeit** lagi.
 - Konfigurasi contoh:
 - Kelas instans DB: Biarkan Kelas standar dipilih.
 - Untuk Kelas instans DB, memilih db.m5.large.
 - Penyimpanan:
 - Mengosongkan kotak Mengaktifkan penskalaan otomatis penyimpanan.
 - Biarkan seluruh pengaturan sebagaimana adanya.
 - Ketersediaan dan Daya Tahan: Tinggalkan Jangan buat instance siaga yang dipilih.
 - Konektivitas:
 - Sumber daya komputasi Tinggalkan Jangan sambungkan ke sumber daya komputasi EC2
 - Jenis jaringan: Biarkan IPv4 dipilih.

- Awan pribadi virtual: DMSVPC-VPC
- Mengakses publik: Ya. Anda harus mengaktifkan dalam mengakses publik untuk menggunakan AWS Schema Conversion Tool.
- Availability zone: us-west-2a
- Biarkan seluruh pengaturan sebagaimana adanya.
- Otentikasi basis data: Biarkan otentikasi Kata Sandi dipilih.
- Di bawah Monitoring, kosongkan kotak Aktifkan Performance Insights. Perluas bagian Konfigurasi tambahan, dan kosongkan kotak Aktifkan Pemantauan yang Ditingkatkan.
- Perluas Konfigurasi tambahan:
 - Di bawah Opsi basis data, masukkan **dms_sample** untuk Nama basis data awal.
 - Di bawah grup parameter DB, pilih dms-mariadb-parameters.
 - Untuk grup Opsi, biarkan default:mariadb-10-6 dipilih.
 - Di bagian Cadangan, lakukan hal berikut:
 - Biarkan Aktifkan pencadangan otomatis dipilih. Basis data sumber Anda harus memiliki pencadangan otomatis yang diaktifkan untuk support replikasi yang sedang berlangsung.
 - Untuk periode retensi Backup, pilih 1 hari.
 - Untuk jendela Backup, biarkan Tidak ada preferensi yang dipilih.
 - Kosongkan kotak Salin tag ke snapshot.
 - Biarkan replikasi Aktifkan di AWS wilayah lain tidak dicentang.
 - Di bawah Enkripsi, mengosongkan kotak Mengaktifkan enkripsi.
 - Biarkan bagian ekspor Log apa adanya.
 - Di bawah Pemeliharaan, kosongkan kotak Aktifkan pemutakhiran versi minor otomatis, dan biarkan pengaturan jendela Pemeliharaan sebagai Tidak ada preferensi.
 - Biarkan Aktifkan perlindungan penghapusan tidak dicentang.

4. Memilih Buat basis data.

Membuat basis data Amazon RDS

Ulangi prosedur sebelumnya untuk membuat basis data Amazon RDS target Anda, dengan perubahan berikut.

Untuk membuat RDS target untuk basis data PostgreSQL

1. Ulangi langkah 1 dan 2 dari prosedur sebelumnya.
2. Pada halaman Buat basis data, mengatur pilihan yang sama, kecuali untuk berikut ini:
 - a. Untuk Opsi mesin, memilih PostgreSQL.
 - b. Untuk Versi, pilih PostgreSQL 13.7-R1
 - c. Untuk Pengidentifikasi instans DB, masukkan **dms-postgresql**.
 - d. Untuk nama pengguna Master, biarkan **postgres** dipilih.
 - e. Untuk grup parameter DB, pilih dms-postgresql-parameters.
 - f. Mengosongkan Mengaktifkan pencadangan otomatis.
3. Memilih Membuat basis data.

Membuat klien Amazon EC2

Dalam bagian ini, Anda membuat klien Amazon EC2. Anda menggunakan klien ini untuk mengisi basis data sumber Anda dengan data untuk mereplikasi. Anda juga menggunakan klien ini untuk memverifikasi replikasi dengan menjalankan kueri pada basis data target.

Menggunakan klien Amazon EC2 untuk mengakses basis data Anda memberikan beberapa keuntungan sebagai berikut dalam mengakses basis data Anda melalui internet:

- Anda dapat membatasi untuk mengakses ke basis data Anda ke klien yang berada di VPC yang sama.
- Kami telah mengkonfirmasi bahwa alat yang Anda gunakan dalam tutorial ini berfungsi, dan mudah dipasang, di Amazon Linux 2023, yang kami rekomendasikan untuk tutorial ini.
- Operasi data antara komponen-komponen dalam VPC umumnya bekerja lebih baik daripada yang melalui internet.

Untuk membuat dan mengkonfigurasi klien Amazon EC2 untuk mengisi basis data sumber Anda

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Pada Dasbor, memilih Meluncurkan instans.
3. Pada halaman Launch an Instance, masukkan nilai berikut:
 - a. Di bagian Nama dan tag, masukkan **DMSClient** Nama.

- b. Di bagian Aplikasi dan Gambar OS (Gambar Mesin Amazon), biarkan pengaturan apa adanya.
- c. Di bagian Jenis Instance, pilih t2.xlarge.
- d. Di bagian Key pair (login), pilih Create a new key pair.
- e. Pada halaman Create key pair, masukkan yang berikut ini:
 - Nama key pair: **DMSKeyPair**
 - Jenis pasangan kunci: Tinggalkan sebagai RSA.
 - Format file kunci pribadi: Pilih pem untuk OpenSSH di macOS atau Linux, atau ppk untuk Putty di Windows.

Simpan file kunci saat diminta.

 Note

Anda juga dapat menggunakan pasangan kunci Amazon EC2 yang ada daripada menciptakan yang baru.

- f. Di bagian Pengaturan Jaringan, pilih Edit. Pilih pengaturan berikut:
 - VPC - diperlukan: Pilih VPC dengan ID yang Anda rekam untuk VPC DMSVPC-VPC.
 - Subnet: Pilih subnet publik pertama.
 - Tetapkan IP publik secara otomatis: Pilih Aktifkan.

Biarkan sisa pengaturan apa adanya, dan pilih Launch instance.

Mengisi basis data sumber Anda

Dalam bagian ini, Anda menemukan titik akhir untuk basis data sumber dan target untuk digunakan kemudian dan menggunakan perangkat berikut untuk mengisi basis data sumber:

- Git, untuk mengunduh penulisan yang mengisi basis data sumber Anda.
- Klien MariaDB, untuk menjalankan skrip ini.

Dapatkan titik akhir

Temukan dan catat titik akhir RDS Anda untuk MariaDB dan RDS untuk instans PostgreSQL DB untuk digunakan nanti.

Menemukan titik akhir instans DB Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di <https://console.aws.amazon.com/rds/>.
2. Pada panel navigasi, memilih Basis data.
3. Pilih database dms-mariadb, dan catat nilai Endpoint untuk database.
4. Ulangi langkah-langkah sebelumnya untuk basis data dms-postgresql.

Mengisi basis data sumber Anda

Selanjutnya, connect ke instans klien Anda, instal perangkat lunak yang diperlukan, mengunduh contoh penulisan basis data AWS dari Git, dan menjalankan penulisan untuk mengisi basis data sumber Anda.

Mengisi basis data sumber Anda

1. Connect ke instans klien menggunakan nama host dan kunci publik yang Anda simpan di langkah-langkah sebelumnya.

Untuk informasi lebih lanjut tentang menghubungkan ke Instans Amazon EC2, lihat [Mengakses Instans](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Note

Jika Anda menggunakan PuTTY, mengaktifkan keepalives TCP pada pengaturan Koneksi sehingga koneksi Anda tidak kehabisan waktu karena tidak mengaktifkan.

2. Instal Git, MariaDB, dan PostgreSQL. Mengonfirmasi penginstalan sesuai kebutuhan.

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

3. Jalankan perintah berikut untuk mengunduh skrip pembuatan database dari GitHub.

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

4. Ubah ke direktori `aws-database-migration-samples/mysql/sampledb/v1/`.
5. Jalankan perintah berikut. Memberikan titik akhir untuk instans RDS sumber yang Anda catat sebelumnya, misalnya `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. Biarkan penulisan pembuatan basis data berjalan. Penulisan memakan waktu hingga 45 menit untuk membuat skema dan mengisi data. Anda dapat dengan aman mengabaikan kesalahan dan peringatan yang ditampilkan skrip.

Lakukan migrasi skema sumber Anda ke basis data target Anda menggunakan AWS SCT

Di bagian ini, Anda menggunakan AWS Schema Conversion Tool untuk melakukan migrasi skema sumber Anda ke basis data target Anda. Atau, Anda dapat menggunakan DMS Schema Conversion untuk mengonversi skema database sumber Anda. Untuk informasi selengkapnya, lihat [Memulai dengan Konversi Skema DMS](#).

Lakukan migrasi skema sumber ke basis data target Anda dengan AWS SCT

1. Instal AWS Schema Conversion Tool. Untuk informasi selengkapnya, lihat [Menginstal, memeriksa, dan memperbarui AWS SCT](#) dalam Panduan Pengguna Schema Conversion Tool AWS.

Ketika Anda mengunduh driver JDBC untuk MySQL dan PostgreSQL, perhatikan di mana Anda menyimpan driver, apabila perangkat ini mengingatkan Anda tentang lokasinya.

2. Buka AWS Schema Conversion Tool. Pilih File, lalu pilih Proyek baru.
3. Dalam jendela Proyek baru, tetapkan nilai-nilai berikut:
 - TETAPKAN Nama Proyek ke **DMSProject**.
 - Jauhkan Lokasi seperti itu untuk menyimpan AWS SCT proyek Anda dalam folder default.

Pilih OKE.

4. Pilih Tambahkan sumber untuk menambahkan database MySQL sumber ke proyek Anda, lalu pilih MySQL, dan pilih Berikutnya.
5. Di halaman Add source, atur nilai berikut:
 - Nama koneksi: **source**
 - Nama Server: Memasukkan titik akhir untuk basis data MySQL yang Anda catat sebelumnya.
 - Port server: **3306**
 - Nama pengguna: **admin**
 - Kata Sandi: **changeit**
6. Pilih Tambahkan target untuk menambahkan target Amazon RDS untuk database PostgreSQL ke proyek Anda, lalu pilih Amazon RDS for PostgreSQL. Pilih Selanjutnya.
7. Dalam halaman Add target, menetapkan nilai-nilai berikut:
 - Nama koneksi: **target**
 - Nama Server: Memasukkan titik akhir untuk basis data MySQL yang Anda catat sebelumnya.
 - Port server: **5432**
 - Database: Masukkan nama database PostgreSQL Anda.
 - Nama pengguna: **postgres**
 - Kata Sandi: **changeit**
8. Di panel kiri, pilih dms_sample di bawah Skema. Di panel kanan, pilih basis data target Anda tentang PostgreSQL. Pilih Buat pemetaan. Anda dapat menambahkan beberapa aturan pemetaan ke satu AWS SCT proyek. Untuk informasi selengkapnya tentang aturan pemetaan, lihat [Membuat aturan pemetaan](#).
9. Pilih Tampilan utama.
10. Di panel kiri, pilih dms_sample di bawah Skema. Buka menu konteks (klik kanan) dan pilih Mengubah skema. Konfirmasikan tindakan.

Setelah perangkat mengubah skema, skema dms_sample muncul di panel kanan.
11. Di panel kanan, di bawah Skema, membuka menu konteks (klik kanan) untuk dms_sample dan memilih Menerapkan ke basis data. Konfirmasikan tindakan.

Verifikasi apakah skema ini selesai. Lakukan langkah-langkah berikut ini.

Untuk memeriksa migrasi skema

1. Connect ke klien Amazon EC2 Anda.
2. Mulai klien PSQL dengan perintah berikut. Tentukan titik akhir basis data PostgreSQL Anda, dan berikan kata sandi basis data bila diminta.

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

3. Kueri salah satu tabel (kosong) untuk memeriksa bahwa AWS SCT menerapkan skema dengan benar,

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

Mengatur replikasi untuk AWS Database Migration Service

Dalam topik ini, Anda mengatur replikasi antara basis data sumber dan target.

Langkah 1: Membuat instans replikasi menggunakan konsol AWS DMS tersebut

Untuk mulai bekerja dengan AWS DMS, pertama-tama membuat instans replikasi.

Instans replikasi melakukan migrasi data aktual antara titik akhir sumber dan target. Instans Anda memerlukan daya penyimpanan dan pemrosesan yang cukup untuk melakukan tugas-tugas yang melakukan migrasi data dari basis data sumber ke basis data target Anda. Seberapa instans replikasi large ini harus tergantung pada jumlah data untuk migrasi dan tugas yang perlu dilakukan instans Anda. Untuk informasi selengkapnya tentang instans replikasi, lihat [Bekerja dengan instance AWS DMS replikasi](#).

DMS > Replication instances > Create replication instance

Create replication instance

Replication instance configuration

Name

The name must be unique among all of your replication instances in the current AWS region.

Type a unique name for your replication instance

Replication instance name must not start with a numeric value

Descriptive Amazon Resource Name (ARN) - optional

A friendly name to override the default DMS ARN. You cannot modify it after creation.

Friendly-ARN-name

Description

Type a short description for your replication instance

The description must only have unicode letters, digits, whitespace, or one of these symbols: _:/=+@. 1000 maximum character.

Instance class [Info](#)

Choose an appropriate instance class for your replication needs. Each instance class provides differing levels of compute, network and memory capacity. [DMS pricing](#)

dms.t2.medium
2 vCPUs 4 GiB Memory

Include previous-generation instance classes

Untuk menciptakan instans replikasi menggunakan konsol tersebut

1. Masuklah ke AWS Management Console dan buka konsol AWS DMS tersebut pada <https://console.aws.amazon.com/dms/v2/>.
2. Pada panel navigasi, pilih Instans replikasi, lalu memilih Buat instans replikasi.
3. Pada halaman Buat instans replikasi, menentukan konfigurasi instans replikasi Anda:
 - a. Untuk Nama, memasukkan **DMS-instance**.
 - b. Untuk Deskripsi, memasukkan deskripsi singkat untuk instans replikasi Anda (opsional).
 - c. Untuk Kelas instans, membiarkan dms.t3.medium dipilih.

Instans membutuhkan penyimpanan, jaringan, dan daya pemrosesan yang cukup untuk migrasi Anda. Untuk informasi selengkapnya bagaimana memilih kelas instans, lihat [Memilih instans replikasi AWS DMS yang tepat untuk migrasi Anda](#).

- d. Untuk Versi mesin, menerima default.
- e. Untuk Multi AZ, pilih Dev atau test workload (Single-AZ).
- f. Untuk Penyimpanan yang dialokasikan (GiB), menerima default 50 GiB.

Dalam AWS DMS, penyimpanan sebagian besar digunakan dengan mencatat file dan transaksi cache. Untuk transaksi cache, penyimpanan hanya digunakan ketika transaksi cache perlu ditulis pada disk. Akibatnya, AWS DMS tidak menggunakan jumlah penyimpanan yang signifikan.

- g. Untuk jenis Jaringan pilih IPv4.
 - h. Untuk VPC, memilih DMSVPC.
 - i. Untuk Grup subnet replikasi, membiarkan grup subnet replikasi saat ini dipilih.
 - j. Jelas Dapat diakses publik.
4. Memilih tab Konfigurasi jaringan dan keamanan lanjutan untuk pengaturan nilai-nilai pengaturan jaringan dan enkripsi jika Anda membutuhkannya:
 - a. Untuk Availability Zone, memilih us-west-2a.
 - b. Untuk Grup keamanan VPC, memilih grup keamanan Default jika belum dipilih.
 - c. Untuk AWS KMS key, biarkan (Default) aws/dms dipilih.
 5. Membiarkan pengaturan pada tab Pemeliharaan sebagaimana adanya. Default adalah jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam per Wilayah AWS, yang dilakukan pada sembarang hari dalam seminggu.
 6. Memilih Membuat.

AWS DMS membuat instans replikasi untuk melakukan migrasi Anda.

Langkah 2: Tentukan titik akhir sumber dan target

Sementara instans replikasi Anda sedang dibuat, Anda dapat menentukan titik akhir untuk menyimpan data sumber dan target untuk basis data Amazon RDS yang Anda buat sebelumnya. Anda membuat setiap titik akhir secara terpisah.

DMS > Endpoints > Create endpoint

Create endpoint

Endpoint type [Info](#)

Source endpoint
A source endpoint allows AWS DMS to read data from a database (on-premises or in the cloud), or from other data source such as Amazon S3.

Target endpoint
A target endpoint allows AWS DMS to write data to a database, or to other data source.

Select RDS DB instance

Endpoint configuration

Endpoint identifier [Info](#)
A label for the endpoint to help you identify it.

ProdEndpoint

Descriptive Amazon Resource Name (ARN) - *optional*
A friendly name to identify the endpoint. You cannot modify it after creation.

Untuk menentukan titik akhir sumber dan titik akhir basis data menggunakan konsol AWS DMS tersebut

1. Pada konsol tersebut, memilih Titik Akhir dari panel navigasi dan kemudian memilih Membuat Titik Akhir.
2. Pada Membuat titik akhir, memilih tipe titik akhir Sumber. Pilih kotak instans Select RDS DB, dan pilih instance dms-mariadb.
3. Dalam bagian Konfigurasi titik akhir, memasukkan **dms-mysql-source** untuk Pengidentifikasi Titik Akhir.
4. Untuk Mesin sumber, membiarkan MySQL dipilih.
5. Untuk Mengakses ke basis data titik akhir, memilih Memberikan informasi untuk mengakses secara manual. Memverifikasi bahwa Port, mode Lapisan Soket Aman (SSL), Nama pengguna, dan Kata Sandi adalah benar.

6. Memilih tab Koneksi titik akhir uji (opsional). Untuk VPC, memilih DMSVPC.
7. Untuk Instans replikasi, membiarkan dms-instance dipilih.
8. Memilih Menjalankan pengujian.

Setelah Anda memilih Jalankan pengujian, AWS DMS menciptakan titik akhir dengan detail yang Anda berikan dan terhubung dengannya. Jika koneksi gagal, edit definisi titik akhir dan uji koneksi lagi. Anda juga dapat menghapus titik akhir secara manual.

9. Setelah Anda berhasil menjalankan pengujian, memilih Membuat titik akhir.
10. Menentukan titik akhir basis data target menggunakan konsol AWS DMS tersebut. Untuk melakukannya, ulangi langkah-langkah sebelumnya, dengan pengaturan berikut:
 - Tipe titik akhir: Titik akhir target
 - Instans RDS: dms-postgresql
 - Identifier titik akhir: **dms-postgresql-target**
 - Mesin target: Biarkan **PostgreSQL** dipilih.

Setelah Anda selesai memberikan semua informasi untuk titik akhir Anda, AWS DMS membuat titik akhir sumber dan target Anda untuk digunakan selama migrasi basis data.

Langkah 3: Buat tugas dan migrasi data

Pada langkah ini, Anda membuat tugas untuk melakukan migrasi data antara basis data yang Anda buat.

DMS > Database migration tasks > Create database migration task

Create database migration task

Task configuration

Task identifier

Replication instance

Source database endpoint

Target database endpoint

Migration type [Info](#)

Untuk membuat tugas migrasi dan mulai migrasi basis data Anda

1. Di panel navigasi konsol tersebut, memilih Tugas migrasi basis data, lalu memilih Membuat tugas. Halaman Membuat tugas migrasi basis data membuka.
2. Dalam bagian Konfigurasi tugas, tentukan pilihan tugas berikut:
 - Pengidentifikasi tugas: Masukkan **dms-task**.
 - <vpc id>Contoh replikasi: Pilih instance replikasi Anda (dms-instance-vpc-).
 - Titik akhir basis data sumber: Pilih dms-mysql-source.
 - Titik akhir basis data target: Pilih dms-postgresql-target.

- Tipe migrasi: Memilih Melakukan migrasi data yang ada dan replikasi perubahan yang sedang berlangsung.
3. Memilih tab Pengaturan tugas. Menetapkan pengaturan berikut:
 - Mode persiapan tabel target: Tidak melakukan apa pun
 - Menghentikan tugas setelah beban penuh selesai: Jangan berhenti
 4. Memilih tab Pemetaan tabel, dan memperluas Aturan Pilihan. Memilih Menambahkan aturan pilihan baru. Menetapkan pengaturan berikut:
 - Skema: Memasukkan skema
 - Nama skema: **dms_sample**
 5. Memilih tab Konfigurasi perusahaan rintisan tugas migrasi, dan kemudian memilih Membuat secara otomatis.
 6. Memilih Membuat tugas.

AWS DMS kemudian membuat tugas migrasi dan mulai. Replikasi basis data awal memakan waktu sekitar 10 menit. Pastikan untuk melakukan langkah berikutnya dalam tutorial sebelum AWS DMS selesai melakukan migrasi data.

Langkah 4: Replikasi pengujian

Pada bagian ini, Anda memasukkan data ke basis data sumber selama dan setelah replikasi awal, dan melakukan kueri basis data target untuk data yang dimasukkan.

Untuk menguji replikasi

1. Pastikan bahwa tugas migrasi basis data Anda menunjukkan status Berjalan tetapi replikasi basis data awal Anda, yang dimulai pada langkah sebelumnya, tidak lengkap.
2. Connect pada klien Amazon EC2 Anda, dan mulai klien MySQL dengan perintah berikut. Menyediakan titik akhir basis data MySQL Anda.

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

3. Jalankan perintah berikut untuk memasukkan catatan ke basis data sumber.

```
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. Keluar dari klien MySQL.

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

5. Sebelum replikasi selesai, lakukan kueri basis data target untuk catatan baru.

Dari instans Amazon EC2, connect pada basis data target menggunakan perintah berikut, memberikan titik akhir basis data target Anda.

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

Berikan kata sandi (**changeit**) saat diminta.

6. Sebelum replikasi selesai, lakukan kueri basis data target untuk catatan baru.

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

7. Ketika tugas migrasi Anda berjalan, Anda dapat memantau kemajuan migrasi basis data Anda sebagaimana yang terjadi:

- Di panel navigasi konsol DMS, memilih Tugas migrasi basis data.
- Memilih dms-tugas.
- Memilih Statistik tabel.

Untuk informasi selengkapnya tentang cara pemantauan pekerjaan, lihat [Pemantauan Tugas AWS DMS](#).

- Setelah replikasi selesai, lakukan kueri basis data target lagi untuk catatan baru. AWS DMS melakukan migrasi catatan baru setelah replikasi awal selesai.

```
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)
```

- Keluar dari klien psql.

```
dms_sample=> quit
```

- Ulangi langkah 1 untuk connect pada basis data sumber lagi.

- Masukkan catatan lain ke dalam tabel 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)
```

- Ulangi langkah 3 dan 4 untuk memutuskan koneksi dari basis data sumber dan connect pada basis data target.
- Kueri basis data target untuk replikasi data lagi.

```
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)
```

Langkah 5: Bersihkan Sumber Daya AWS DMS

Setelah menyelesaikan tutorial memulai, Anda dapat menghapus sumber daya yang Anda buat. Anda dapat menggunakan konsol AWS tersebut untuk menghapusnya. Pastikan untuk menghapus tugas migrasi sebelum menghapus instans replikasi dan titik akhir.

Untuk menghapus tugas migrasi menggunakan konsol tersebut

- Pada panel navigasi konsol AWS DMS tersebut, memilih Tugas migrasi basis data.

2. Memilih dms-tugas.
3. Memilih Tindakan, Menghapus.

Untuk menghapus instans replikasi menggunakan konsol tersebut

1. Pada panel navigasi konsol AWS DMS tersebut, memilih Instans replikasi.
2. Memilih DMS-instans.
3. Memilih Tindakan, Menghapus.

AWS DMS menghapus instans replikasi dan membersihkannya dari halaman Instans replikasi.

Untuk menghapus titik akhir menggunakan konsol tersebut

1. Pada panel navigasi konsol AWS DMS tersebut, memilih Titik akhir.
2. Pilih dms-mysql-source.
3. Memilih Tindakan, Menghapus.

Setelah menghapus sumber daya AWS DMS, pastikan juga untuk menghapus sumber daya berikut.

Untuk bantuan menghapus sumber daya di layanan lainnya, lihat dokumentasi setiap layanan.

- Basis data RDS Anda.
- Grup parameter basis data RDS Anda.
- Grup subnet RDS Anda.
- CloudWatch Log Amazon apa pun yang dibuat bersama dengan database dan instance replikasi Anda.
- Grup keamanan yang dibuat untuk klien Amazon VPC dan Amazon EC2 Anda. Pastikan untuk menghapus aturan masuk dari Default untuk grup keamanan Launch Wizard-1, yang diperlukan bagi Anda untuk dapat menghapusnya.
- Klien Amazon EC2 Anda.
- Amazon VPC Anda.
- Pasangan kunci Amazon EC2 Anda untuk klien Amazon EC2 Anda.

Sumber daya tambahan untuk bekerja dengan AWS Database Migration Service

Kemudian dalam panduan ini, Anda bisa belajar cara menggunakan AWS DMS untuk melakukan migrasi data Anda ke dan dari basis data komersial dan sumber terbuka yang paling banyak digunakan.

Kami juga menyarankan agar Anda memeriksa sumber daya berikut saat Anda mempersiapkan dan melakukan proyek migrasi basis data:

- [AWS DMS Panduan](#) ini memberikan step-by-step panduan yang melalui proses migrasi data ke AWS.
- [AWS DMS Referensi API](#) — Referensi ini menjelaskan semua operasi API untuk AWS Database Migration Service secara detail.
- [AWS CLI untuk AWS DMS](#) - Referensi ini memberikan informasi tentang menggunakan AWS Command Line Interface (AWS CLI) dengan AWS DMS.

Menemukan dan mengevaluasi database untuk migrasi dengan Fleet Advisor AWS DMS

Anda dapat menggunakan DMS Fleet Advisor untuk mengumpulkan metadata dan metrik kinerja dari beberapa lingkungan database. Metrik yang dikumpulkan ini memberikan wawasan tentang infrastruktur data Anda. [DMS Fleet Advisor](#) mengumpulkan metadata dan metrik dari database lokal dan server analitik Anda dari satu atau beberapa lokasi pusat tanpa perlu menginstalnya di setiap komputer. Saat ini, DMS Fleet Advisor mendukung penemuan dan pengumpulan metrik untuk server database Microsoft SQL Server, MySQL, Oracle, dan PostgreSQL.

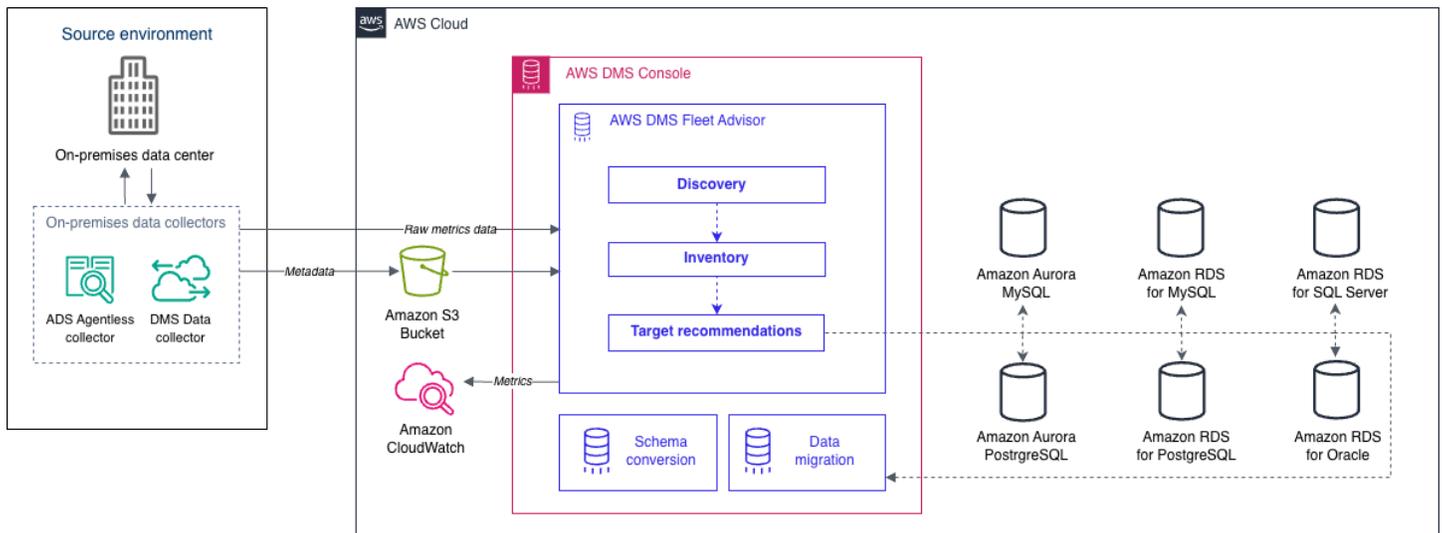
Berdasarkan data yang ditemukan dari jaringan, Anda dapat membangun inventaris untuk menentukan daftar server database untuk pengumpulan data lebih lanjut. Setelah AWS DMS mengumpulkan informasi tentang server, database, dan skema Anda, Anda dapat menganalisis kelayakan migrasi database yang Anda inginkan.

Untuk database dalam inventaris yang Anda rencanakan untuk dimigrasi AWS Cloud, DMS Fleet Advisor menghasilkan rekomendasi target berukuran tepat. Untuk menghasilkan rekomendasi target, DMS Fleet Advisor mempertimbangkan metrik dari pengumpul data dan pengaturan pilihan Anda. Setelah DMS Fleet Advisor menghasilkan rekomendasi, Anda dapat melihat informasi terperinci untuk setiap konfigurasi basis data target. Insinyur dan administrator database organisasi Anda dapat menggunakan Rekomendasi Target Penasihat Armada DMS untuk merencanakan migrasi basis data lokal mereka. AWS Anda dapat menjelajahi berbagai opsi migrasi yang tersedia dan mengeksport rekomendasi ini ke dalam AWS Pricing Calculator untuk lebih mengoptimalkan biaya.

Untuk daftar database sumber yang didukung, lihat [Sumber untuk DMS Fleet Advisor](#).

Untuk daftar database yang digunakan DMS Fleet Advisor untuk menghasilkan rekomendasi target, lihat [Target untuk DMS Fleet Advisor](#). DMS Fleet Advisor menghasilkan like to like recommendations, misalnya, dari sumber Oracle untuk menargetkan database Oracle. DMS Fleet Advisor juga menghasilkan rekomendasi heterogen, seperti migrasi dari sumber Oracle atau Microsoft SQL Server untuk menargetkan RDS untuk PostgreSQL atau Aurora PostgreSQL database.

Diagram berikut menggambarkan proses Rekomendasi Target Penasihat AWS DMS Armada.



Gunakan topik berikut untuk lebih memahami cara menggunakan AWS DMS Fleet Advisor.

Topik

- [Wilayah AWS yang Didukung](#)
- [Memulai dengan Fleet Advisor](#)
- [Menyiapkan Penasihat AWS DMS Armada](#)
- [Menemukan database untuk migrasi menggunakan pengumpul data](#)
- [Menggunakan inventaris untuk analisis di AWS DMS Fleet Advisor](#)
- [Menggunakan fitur Rekomendasi Target Penasihat AWS DMS Armada](#)
- [Batasan Penasihat Armada DMS](#)

Wilayah AWS yang Didukung

Anda dapat menggunakan DMS Fleet Advisor berikut ini. Wilayah AWS

Nama Wilayah	Wilayah
AS Timur (Virginia Utara)	us-east-1
US East (Ohio)	us-east-2
US West (Northern California)	us-west-1

Nama Wilayah	Wilayah
US West (Oregon)	us-west-2
Asia Pasifik (Hong Kong)	ap-east-1
Asia Pasifik (Tokyo)	ap-northeast-1
Asia Pasifik (Seoul)	ap-northeast-2
Asia Pacific (Osaka)	ap-northeast-3
Asia Pasifik (Mumbai)	ap-south-1
Asia Pasifik (Singapura)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pasifik (Jakarta)	ap-southeast-3
Kanada (Pusat)	ca-central-1
Europe (Frankfurt)	eu-central-1
Eropa (Stockholm)	eu-north-1
Eropa (Irlandia)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Eropa (Milan)	eu-selatan-3
Kanada (Pusat)	ca-central-1
Amerika Selatan (Sao Paulo)	sa-east-1
Timur Tengah (Bahrain)	me-south-1
Afrika (Cape Town)	af-south-1

Memulai dengan Fleet Advisor

Anda dapat menggunakan DMS Fleet Advisor untuk menemukan database lokal sumber untuk migrasi keAWS Cloud. Kemudian, Anda dapat menentukan target migrasi yang tepat diAWS Cloud untuk setiap database lokal Anda. Gunakan alur kerja berikut untuk membuat inventaris database sumber Anda dan menghasilkan rekomendasi target.

1. Buat bucket Amazon S3, kebijakan IAM, peran, dan pengguna. Untuk informasi selengkapnya, lihat [Membuat sumber daya yang dibutuhkan](#).
2. Buat pengguna database dengan izin minimum yang diperlukan untuk pengumpul data DMS. Untuk informasi selengkapnya, lihat [Membuat pengguna database](#).
3. Buat dan unduh pengumpul data. Untuk informasi selengkapnya, lihat [Membuat pengumpul data](#).
4. Mulai menggunakan pengumpul data di lingkungan lokal Anda. Selanjutnya, konfigurasi pengumpul data Anda untuk memastikannya dapat mengirim data yang dikumpulkan ke DMS Fleet Advisor. Untuk informasi selengkapnya, lihat [Memasang pengumpul data](#).
5. Temukan OS dan server database di lingkungan data Anda. Untuk informasi selengkapnya, lihat [Menemukan server OS dan database](#).
6. Kumpulkan metadata database dan metrik pemanfaatan sumber daya. Untuk informasi selengkapnya, lihat [Mengumpulkan data](#).
7. Analisis basis data dan skema sumber Anda. DMS Fleet Advisor menjalankan penilaian skala besar database Anda untuk mengidentifikasi skema serupa. Untuk informasi selengkapnya, lihat [Menggunakan inventaris untuk analisis di AWS DMS Fleet Advisor](#).
8. Buat, lihat, dan simpan salinan lokal rekomendasi target untuk database sumber Anda. Untuk informasi selengkapnya, lihat [Rekomendasi target](#).

Setelah Anda menentukan target migrasi untuk setiap database sumber, Anda dapat menggunakan Konversi Skema DMS untuk mengonversi skema database Anda ke platform baru. Kemudian, Anda dapat menggunakanAWS DMS untuk memigrasi data. Untuk informasi lebih lanjut, lihat[Mengonversi skema database menggunakan Konversi Skema DMS](#) dan[Apa itu AWS Database Migration Service?](#)

[Video ini](#) memperkenalkan antarmuka pengguna Konversi Skema DMS dan membantu Anda membiasakan diri dengan komponen inti dari layanan ini.

Menyiapkan Penasihat AWS DMS Armada

Untuk mengatur Penasihat AWS DMS Armada, selesaikan tugas prasyarat berikut.

Topik

- [Membuat AWS sumber daya yang dibutuhkan untuk AWS DMS Fleet Advisor](#)
- [Membuat pengguna database untuk AWS DMS Fleet Advisor](#)

Membuat AWS sumber daya yang dibutuhkan untuk AWS DMS Fleet Advisor

DMS Fleet Advisor membutuhkan satu set AWS sumber daya di akun Anda untuk meneruskan dan mengimpor informasi inventaris, dan untuk memperbarui status pengumpul data DMS.

Sebelum Anda mengumpulkan data dan membuat inventaris database dan skema untuk pertama kalinya, selesaikan prasyarat berikut.

Untuk mengonfigurasi bucket Amazon S3 dan sumber daya IAM, lakukan salah satu hal berikut:

- [Konfigurasi sumber daya Amazon S3 dan IAM menggunakan AWS CloudFormation](#) (direkomendasikan).
- [Konfigurasi sumber daya Amazon S3 dan IAM di AWS Management Console](#)

Konfigurasi sumber daya Amazon S3 dan IAM menggunakan AWS CloudFormation

CloudFormation Tumpukan adalah kumpulan sumber AWS daya yang dapat Anda kelola sebagai satu unit. Untuk menyederhanakan pembuatan sumber daya yang diperlukan untuk DMS Fleet Advisor, Anda dapat menggunakan file AWS CloudFormation template untuk membuat tumpukan. CloudFormation Untuk informasi selengkapnya, lihat [Membuat tumpukan di AWS CloudFormation konsol](#) di Panduan AWS CloudFormation Pengguna.

Note

Bagian ini hanya berlaku untuk menggunakan kolektor DMS Fleet Advisor mandiri. [Untuk informasi tentang menggunakan kolektor lokal tunggal untuk mengumpulkan informasi tentang database dan server, lihat Application Discovery Service Agentless Collector di Panduan Pengguna. AWS Application Discovery Service](#)

Sumber daya Amazon S3 dan IAM dibuat oleh CloudFormation

Saat Anda menggunakan CloudFormation templat, mereka membuat tumpukan yang menyertakan sumber daya berikut di: Akun AWS

- Bucket Amazon S3 bernama `dms-fleetadvisor-data-accountId-region`
- Pengguna IAM bernama `FleetAdvisorCollectorUser-region`
- Peran layanan IAM bernama `FleetAdvisorS3Role-region`
- Kebijakan akses bernama `FleetAdvisorS3Role-region-Policy`
- Kebijakan akses bernama `FleetAdvisorCollectorUser-region-Policy`
- Peran Tertaut Layanan IAM (SLR) bernama `AWSServiceRoleForDMSFleetAdvisor`

Ikuti langkah-langkah yang tercantum di bawah ini untuk mengonfigurasi sumber daya Anda CloudFormation.

- [Langkah 1: Unduh file CloudFormation template](#)
- [Langkah 2: Konfigurasi Amazon S3 dan IAM menggunakan CloudFormation](#)

Langkah 1: Unduh file CloudFormation template

CloudFormation Template adalah deklarasi AWS sumber daya yang membentuk tumpukan. Templat ini disimpan sebagai file JSON.

Untuk mengunduh file CloudFormation template

1. Buka menu konteks (klik kanan) untuk salah satu tautan berikut dan pilih Simpan Tautan Sebagai:
 - [Jika Anda berencana untuk menggunakan DMS Fleet Advisor, pilih `dms-fleetadvisor-iam-slr-s3.zip`. Jika Anda telah membuat SLR untuk DMS Fleet Advisor, pilih `3.zip dms-fleetadvisor-iam-s`](#)
 - [Jika Anda berencana untuk menggunakan AWS Application Discovery Service \(ADS\) Agentless Collector dan belum membuat SLR untuk itu, maka pilih `-slr-s3.zip. dms-fleetadvisor-ads-iam`. Jika Anda telah membuat SLR untuk DMS Fleet Advisor dengan ADS sebelumnya, pilih `-s3.zip. dms-fleetadvisor-ads-iam`](#)
2. Simpan file tersebut ke komputer Anda.

Langkah 2: Konfigurasi Amazon S3 dan IAM menggunakan CloudFormation

Saat Anda menggunakan CloudFormation templat untuk IAM, templat ini akan membuat sumber daya Amazon S3 dan IAM yang terdaftar sebelumnya.

Untuk mengkonfigurasi Amazon S3 dan IAM menggunakan CloudFormation

1. Buka CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Mulai wizard Create Stack dengan memilih Create Stack dan With new resources di daftar dropdown.
3. Pada halaman Buat tumpukan, lakukan hal berikut:
 - a. Untuk Siapkan templat, pilih Templat sudah siap.
 - b. Untuk Sumber templat, pilih Unggah file templat.
 - c. Untuk Pilih file, navigasikan ke, lalu pilih `dms-fleetadvisor-iam-slr-s3.json`, `-s3.json`. `dms-fleetadvisor-iam`, `dms-fleetadvisor-ads-iam-slr-s3.zip`, atau `dms-fleetadvisor-ads-iam-s3.zip`.
 - d. Pilih Berikutnya.
4. Pada halaman Tentukan detail tumpukan, lakukan hal berikut:
 - a. Untuk nama Stack, masukkan `dms-fleetadvisor-iam-slr-s3`, `dms-fleetadvisor-iam-s3`, `dms-fleetadvisor-ads-iam-slr-s3`, atau `dms-fleetadvisor-ads-iam-s3`.
 - b. Pilih Berikutnya.
5. Pada halaman Konfigurasi opsi tumpukan, pilih Berikutnya.
6. Pada halaman Review `dms-fleetadvisor-iam-slr-s3`, Review `dms-fleetadvisor-iam-s3`, Review `dms-fleetadvisor-ads-iam-slr-s3`, atau Review `dms-fleetadvisor-ads-iam-s3`, lakukan hal berikut:
 - a. Pilih kotak centang Saya memahami bahwa AWS CloudFormation dapat membuat sumber daya IAM dengan nama kustom.
 - b. Pilih Kirim.

CloudFormation membuat bucket S3 dan peran IAM serta pengguna yang dibutuhkan DMS Fleet Advisor. Di panel kiri, ketika `dms-fleetadvisor-iam-slr-s3`, `3`, `dms-fleetadvisor-iam-sdms-fleetadvisor-ads-iam-slr-s3`, atau `dms-fleetadvisor-ads-iam-s3` menunjukkan `CREATE_COMPLETE`, lanjutkan ke langkah berikutnya.

7. Di panel kiri, pilih `dms-fleetadvisor-iam-slr-s3`, `3`, `dms-fleetadvisor-iam-sdms-fleetadvisor-ads-iam-slr-s3`, atau `-s3`. `dms-fleetadvisor-ads-iam` Di panel kanan, lakukan hal berikut:
 - a. Pilih Info tumpukan. ***Tumpukan Anda memiliki ID dalam 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`, atau `arn:aws:cloudformation: region: account-not-no:: tumpukan/-s3/pengidentifikasi dms-fleetadvisor-iam-s dms-fleetadvisor-ads-iam dms-fleetadvisor-ads-iam`.***
 - b. Pilih Sumber daya. Anda akan melihat yang berikut ini:
 - Bucket Amazon S3 bernama `dms-fleetadvisor-data-accountId-region`
 - Peran layanan bernama `FleetAdvisorS3Role-region`
 - Pengguna IAM bernama `FleetAdvisorCollectorUser-region`
 - SLR IAM bernama `AWSServiceRoleForDMSFleetAdvisor` (jika Anda mengunduh `dms-fleet-advisor-iam-slr-s3.zip` atau `dms-fleet-advisor-ads-iam-slr-s3.zip`).
 - Kebijakan akses bernama `FleetAdvisorS3Role-region-Policy`
 - Kebijakan akses bernama `FleetAdvisorCollectorUser-region-Policy`

Konfigurasi sumber daya Amazon S3 dan IAM di AWS Management Console

Buat bucket Amazon S3.

Buat bucket Amazon S3 tempat metadata inventaris dapat disimpan. Kami menyarankan Anda mengkonfigurasi bucket S3 ini sebelum menggunakan DMS Fleet Advisor. AWS DMS menyimpan metadata inventaris DMS Fleet Advisor Anda di bucket S3 ini.

Untuk informasi selengkapnya tentang membuat bucket S3, lihat [Membuat bucket S3 pertama Anda](#) di Panduan Pengguna Amazon S3.

Note

DMS Fleet Advisor hanya mendukung bucket terenkripsi SSE-S3.

Untuk membuat bucket Amazon S3 untuk menyimpan informasi lingkungan data lokal

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>.
2. Pilih Buat bucket.
3. *Di halaman **Buat bucket**, masukkan nama unik global yang menyertakan nama login untuk bucket, seperti **fa-bucket- yoursignin**.*
4. Pilih Wilayah AWS tempat Anda menggunakan DMS Fleet Advisor.
5. Simpan pengaturan yang tersisa dan pilih Buat ember.

Buat sumber daya IAM

Di bagian ini, Anda membuat sumber daya IAM untuk pengumpul data, pengguna IAM, dan Penasihat Armada DMS.

Topik

- [Buat sumber daya IAM untuk pengumpul data Anda](#)
- [Buat peran terkait layanan DMS Fleet Advisor](#)

Buat sumber daya IAM untuk pengumpul data Anda

Untuk memastikan pengumpul data berfungsi dengan benar dan mengunggah metadata yang dikumpulkan ke bucket Amazon S3, buat kebijakan berikut. Kemudian, buat pengguna IAM dengan izin minimum berikut. Untuk informasi selengkapnya tentang pengumpul data DMS, lihat [Menemukan database untuk migrasi menggunakan pengumpul data](#).

Untuk membuat kebijakan IAM untuk DMS Fleet Advisor dan pengumpul data Anda untuk mengakses Amazon S3

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat kebijakan.
4. Di halaman Buat kebijakan, pilih tab JSON.
5. Tempelkan JSON berikut ke editor, ganti kode contoh. Ganti *fa_bucket* dengan nama bucket Amazon S3 yang Anda buat di bagian sebelumnya.

```
{
  "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. Pilih Berikutnya: Tag dan Berikutnya: Tinjau.
7. Masukkan **FleetAdvisorS3Policy** Nama*, lalu pilih Buat kebijakan.

Untuk membuat kebijakan IAM bagi pengumpul data DMS untuk mengakses DMS Fleet Advisor

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat kebijakan.
4. Di halaman Buat kebijakan, pilih tab JSON.
5. Tempelkan kode JSON berikut ke editor, ganti kode contoh.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dms:DescribeFleetAdvisorCollectors",

```

```

        "dms:ModifyFleetAdvisorCollectorStatuses",
        "dms:UploadFileMetadataList"
    ],
    "Resource": "*"
}
]
}

```

6. Pilih Berikutnya: Tag dan Berikutnya: Tinjau.
7. Masukkan **DMSCollectorPolicy** Nama*, lalu pilih Buat kebijakan.

Untuk membuat pengguna IAM dengan izin minimum untuk menggunakan pengumpul data DMS

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Users (Pengguna).
3. Pilih Add Users (Tambahkan pengguna).
4. Pada halaman Tambah pengguna, masukkan **FleetAdvisorCollectorUser** nama pengguna*. Pilih Access key- Programmatic Access untuk Pilih Jenis AWS Akses. Pilih Selanjutnya: Izin.
5. Di bagian Setel izin, pilih Lampirkan kebijakan yang ada secara langsung.
6. Gunakan kontrol pencarian untuk menemukan dan memilih kebijakan DMS CollectorPolicy dan FleetAdvisorS3Policy yang Anda buat sebelumnya. Pilih Berikutnya: Tanda.
7. Pada halaman Tag, pilih Next: Review.
8. Pada halaman Tinjauan, pilih Buat pengguna. Di halaman berikutnya, pilih Unduh.csv untuk menyimpan kredensial pengguna baru. Gunakan kredensial ini dengan DMS Fleet Advisor untuk izin akses minimum yang diperlukan.

Untuk membuat peran IAM untuk DMS Fleet Advisor dan pengumpul data Anda untuk mengakses Amazon S3

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, silakan pilih Peran.
3. Pilih Buat peran.

4. Pada halaman Pilih entitas tepercaya, untuk jenis entitas tepercaya, pilih AWSLayanan. Untuk kasus penggunaan untuk AWS layanan lain, pilih DMS.
5. Pilih kotak centang DMS dan pilih Berikutnya.
6. Pada halaman Tambahkan izin, pilih FleetAdvisorS3Policy. Pilih Berikutnya.
7. Pada halaman Nama, tinjau, dan buat, masukkan **FleetAdvisorS3Role** nama Peran, lalu pilih Buat peran.
8. Pada halaman Peran, masukkan nama **FleetAdvisorS3Role** Peran. Pilih FleetAdvisorS3Role.
9. Pada halaman FleetAdvisorS3Role, pilih tab Trust relationship. Pilih Edit kebijakan kepercayaan.
10. Pada halaman Edit kebijakan kepercayaan, tempelkan JSON berikut ke editor, ganti teks yang ada.

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

Kebijakan sebelumnya memberikan `sts:AssumeRole` izin ke layanan yang AWS DMS digunakan untuk mengimpor data yang dikumpulkan dari bucket Amazon S3.

11. Pilih Perbarui kebijakan.

Buat peran terkait layanan DMS Fleet Advisor

DMS Fleet Advisor menggunakan peran terkait layanan untuk mengelola metrik Amazon CloudWatch di Anda. Akun AWS DMS Fleet Advisor menggunakan peran terkait layanan ini untuk mempublikasikan metrik kinerja database yang dikumpulkan atas nama Anda. CloudWatch

Untuk membuat peran terkait layanan untuk DMS Fleet Advisor

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, silakan pilih Peran. Kemudian, pilih Buat peran.
3. Untuk jenis entitas Tepercaya, pilih AWSlayanan.
4. Untuk kasus Penggunaan untuk AWS layanan lain, pilih DMS — Fleet Advisor.
5. Pilih kotak centang DMS - Fleet Advisor dan pilih Berikutnya.
6. Pada halaman Tambahkan izin, pilih Berikutnya.
7. Pada halaman Nama, tinjau, dan buat, pilih Buat peran.

Atau, Anda dapat membuat peran terkait layanan ini dari AWS API atau CLIAWS. Untuk informasi selengkapnya, lihat [Membuat peran terkait layanan untuk AWS DMS Fleet Advisor](#).

Setelah membuat peran terkait layanan untuk DMS Fleet Advisor, Anda dapat melihat metrik kinerja untuk basis data sumber Anda dalam rekomendasi target. Selain itu, Anda dapat melihat metrik ini dan di CloudWatch akun Anda. Untuk informasi selengkapnya, lihat [Rekomendasi target](#).

Untuk membuat kebijakan IAM yang diperlukan untuk peran terkait layanan DMS Fleet Advisor

Izin minimum yang diperlukan untuk membuat peran terkait layanan ditentukan dalam kebijakan. `DMSFleetAdvisorCreateServiceLinkedRolePolicy` Buat kebijakan IAM ini untuk akun Anda jika Anda tidak dapat membuat peran terkait layanan.

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat kebijakan.
4. Di halaman Buat kebijakan, pilih tab JSON.
5. Tempelkan kode JSON berikut ke editor, ganti kode contoh.

```
{
  "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. Pilih Berikutnya: Tag dan Berikutnya: Tinjau.
7. Masukkan **DMSFleetAdvisorCreateServiceLinkedRolePolicy** Nama*, lalu pilih Buat kebijakan.

Sekarang, Anda dapat menggunakan kebijakan ini untuk membuat peran terkait layanan untuk DMS Fleet Advisor.

Membuat pengguna database untuk AWS DMS Fleet Advisor

Bagian ini menjelaskan cara membuat pengguna untuk database sumber Anda dengan izin minimum yang diperlukan untuk pengumpul data DMS.

Bagian ini berisi topik berikut:

- [Menggunakan pengguna database dengan AWS DMS Fleet Advisor](#)
- [Membuat pengguna database dengan MySQL](#)
- [Membuat pengguna database dengan Oracle](#)
- [Membuat pengguna database dengan PostgreSQL](#)
- [Membuat pengguna database dengan Microsoft SQL Server](#)
- [Menghapus pengguna database](#)

Menggunakan pengguna database dengan AWS DMS Fleet Advisor

Anda dapat menggunakan pengguna database selain root dengan pengumpul data DMS. Tentukan nama pengguna dan kata sandi setelah menambahkan database ke inventaris, tetapi sebelum Anda menjalankan pengumpul data Anda. Untuk informasi selengkapnya tentang menambahkan database ke inventaris, lihat [Mengelola objek yang dipantau](#).

Setelah Anda selesai menggunakan pengumpul data DMS, Anda dapat menghapus pengguna database yang Anda buat. Untuk informasi selengkapnya, lihat [Menghapus pengguna database](#).

Important

Dalam contoh berikut, ganti `{your_user_name}` dengan nama pengguna database yang Anda buat untuk database Anda. Kemudian, ganti `{your_password}` dengan kata sandi yang aman.

Membuat pengguna database dengan MySQL

Untuk membuat pengguna database dalam database sumber MySQL, gunakan skrip berikut. Pastikan Anda menyimpan satu versi GRANT pernyataan yang bergantung pada versi database MySQL Anda.

```
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;

```

Membuat pengguna database dengan Oracle

Untuk membuat pengguna database dalam database sumber Oracle, gunakan skrip berikut.

Untuk menjalankan skrip SQL ini, sambungkan ke database Oracle Anda menggunakan SYSDBA hak istimewa. Setelah Anda menjalankan skrip SQL ini, sambungkan ke database Anda menggunakan kredensial pengguna yang Anda buat dengan skrip ini. Juga, gunakan kredensial pengguna ini untuk menjalankan pengumpul data DMS.

Script berikut menambahkan C## awalan untuk nama pengguna untuk Oracle multitenant database container (CDB).

```

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;

```

Membuat pengguna database dengan PostgreSQL

Untuk membuat pengguna database dalam database sumber PostgreSQL, gunakan skrip berikut.

```

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}";

```

Membuat pengguna database dengan Microsoft SQL Server

Untuk membuat pengguna database dalam database sumber Microsoft SQL Server, gunakan skrip berikut.

```

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}]
```

Menghapus pengguna database

Setelah Anda menyelesaikan semua tugas pengumpulan data, Anda dapat menghapus pengguna database yang Anda buat untuk pengumpul data DMS. Anda dapat menggunakan skrip berikut untuk menghapus pengguna dengan izin minimum dari database Anda.

Untuk menghapus pengguna dari database MySQL Anda, jalankan skrip berikut.

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

Untuk menghapus pengguna dari database Oracle Anda, jalankan skrip berikut.

```
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;
```

Untuk menghapus pengguna dari database PostgreSQL Anda, jalankan skrip berikut.

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

Untuk menghapus pengguna dari database SQL Server Anda, jalankan skrip berikut.

```
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
```

Menemukan database untuk migrasi menggunakan pengumpul data

Untuk menemukan infrastruktur data sumber, Anda dapat menggunakan [AWS Application Discovery Service Agentless Collector](#) atau [pengumpul AWS DMS data](#). ADS Agentless Collector adalah aplikasi lokal yang mengumpulkan informasi tentang lingkungan lokal Anda melalui metode tanpa agen, termasuk informasi profil server (misalnya, OS, jumlah CPU, jumlah RAM), metadata database, dan metrik pemanfaatan. Anda menginstal Agentless Collector sebagai mesin virtual (VM) di lingkungan VMware vCenter Server Anda menggunakan file Open Virtualization Archive (OVA). Pengumpul AWS DMS data adalah aplikasi Windows yang Anda instal di lingkungan lokal Anda. Aplikasi ini terhubung ke lingkungan data Anda dan mengumpulkan metadata dan metrik kinerja dari database lokal dan server analitik Anda. Setelah metadata database dan metrik kinerja dikumpulkan baik melalui ADS Agentless Collector atau DMS data collector, DMS Fleet Advisor membangun inventaris server, database, dan skema yang dapat Anda migrasi ke AWS Cloud.

Pengumpul data DMS adalah aplikasi Windows yang menggunakan pustaka, konektor, dan penyedia data .NET untuk terhubung ke database sumber Anda untuk penemuan database dan pengumpulan data.

Pengumpul data DMS berjalan pada Windows. Namun, pengumpul data DMS Anda dapat mengumpulkan data dari semua vendor database yang didukung terlepas dari server OS tempat mereka berjalan.

Pengumpul data DMS menggunakan protokol RTPS yang dilindungi dengan enkripsi TLS untuk membuat koneksi aman dengan DMS Fleet Advisor. Oleh karena itu, data Anda dienkripsi selama transit secara default.

AWS DMS memiliki jumlah maksimum pengumpul data yang dapat Anda buat untuk Anda Akun AWS. Lihat bagian berikut untuk informasi tentang kuota AWS DMS [Kuota untuk AWS Database Migration Service](#) layanan.

Topik

- [Izin untuk pengumpul data DMS](#)
- [Membuat pengumpul data untuk AWS DMS Fleet Advisor](#)
- [Menginstal dan mengkonfigurasi pengumpul data](#)
- [Menemukan server OS dan database untuk dipantau](#)
- [Mengelola objek yang dipantau](#)
- [Menggunakan SSL dengan AWS DMS Fleet Advisor](#)
- [Mengumpulkan data untuk AWS DMS Fleet Advisor](#)
- [Pemecahan masalah untuk pengumpul data DMS](#)

Izin untuk pengumpul data DMS

Pengguna database yang Anda buat untuk pengumpul data DMS harus memiliki izin baca. Namun, dalam beberapa kasus, pengguna database memerlukan EXECUTE izin. Untuk informasi selengkapnya, lihat [Membuat pengguna database untuk AWS DMS Fleet Advisor](#).

Pengumpul data DMS memerlukan izin tambahan untuk menjalankan skrip penemuan.

- Untuk penemuan OS, pengumpul data DMS membutuhkan kredensial untuk server domain untuk menjalankan permintaan menggunakan protokol LDAP.
- Untuk penemuan database di Linux, pengumpul data DMS membutuhkan kredensial dengan hibah. `sudo SSH` Juga, Anda harus mengkonfigurasi server Linux Anda untuk memungkinkan menjalankan skrip SSH jarak jauh.
- Untuk penemuan database di Windows, pengumpul data DMS membutuhkan kredensial dengan hibah untuk menjalankan kueri Windows Management Instrumentation (WMI) dan WMI Query Language (WQL) dan membaca registri. Selain itu, Anda harus mengonfigurasi server Windows Anda untuk memungkinkan menjalankan WMI, WQL, dan skrip jarak jauh. PowerShell

Membuat pengumpul data untuk AWS DMS Fleet Advisor

Pelajari cara membuat dan mengunduh pengumpul data DMS.

Sebelum Anda membuat pengumpul data, gunakan konsol IAM untuk membuat peran terkait layanan untuk DMS Fleet Advisor. Peran ini memungkinkan kepala sekolah untuk mempublikasikan titik data metrik ke Amazon. CloudWatch DMS Fleet Advisor menggunakan peran ini untuk menampilkan bagan dengan metrik database. Untuk informasi selengkapnya, lihat [Membuat peran terkait layanan untuk AWS DMS Fleet Advisor](#).

Untuk membuat dan mengunduh pengumpul data

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.

Pilih Wilayah tempat Anda menggunakan DMS Fleet Advisor.

2. Di panel navigasi, pilih Pengumpul data di bawah Temukan. Halaman pengumpul data terbuka.
3. Pilih Buat pengumpul data. Halaman Buat pengumpul data terbuka.

DMS > Discover: Data collectors > Create data collector

Create data collector [Info](#)

Create a data collector to identify servers, databases, and schemas on a network. After the data collector is created, you're prompted to register it by downloading and installing a local collector.

 You can create a maximum of 10 data collectors. [Learn more](#) 

General configuration

Name

Can have only Unicode letters, digits, white space, or one of the symbols in parentheses: `[_:/=+-@()]`. Maximum of 60 characters.

Description - optional

Provide a description of the data collector purpose, environment, or network to help you identify it in the future.

Can have only Unicode letters, digits, white space, or one of the symbols in parentheses: `[_:/=+-@()]`. Maximum of 255 characters.

Connectivity [Info](#)

Amazon S3 bucket

Choose or create an Amazon S3 bucket to store collected metadata. Ensure this bucket is the currently selected region.

[View](#) [Browse S3](#)

To create a bucket role, go to [S3](#) 

IAM role

Choose or create an IAM role that grants AWS DMS permissions to access the specified S3 bucket.

To create an IAM role, go to [IAM console](#) 

[Cancel](#)[Create data collector](#)

4. Untuk Nama di bagian Konfigurasi umum, masukkan nama pengumpul data Anda.

5. Di bagian Konektivitas, pilih Browse S3. Pilih bucket Amazon S3 yang telah Anda konfigurasi sebelumnya dari daftar yang muncul.

AWS DMS menyimpan metadata inventaris DMS Fleet Advisor Anda di bucket S3 ini. Pastikan bucket Amazon S3 Anda berada di Wilayah AWS tempat yang sama dengan AWS DMS Fleet Advisor Anda saat ini berjalan.

 Note

DMS Fleet Advisor hanya mendukung bucket terenkripsi SSE-S3.

6. Dalam daftar peran IAM, pilih peran IAM yang telah Anda konfigurasi sebelumnya dari daftar yang muncul. Peran ini memberikan AWS DMS izin untuk mengakses bucket Amazon S3 yang ditentukan.
7. Pilih Buat pengumpul data. Halaman pengumpul data terbuka dan pengumpul data yang dibuat muncul dalam daftar.

Saat membuat pengumpul data pertama Anda, AWS DMS mengonfigurasi lingkungan di bucket Amazon S3 Anda yang memformat data dan menyimpan atribut untuk digunakan dengan DMS Fleet Advisor.

8. Pilih Unduh kolektor lokal di spanduk informasi untuk mengunduh pengumpul data yang baru Anda buat. Sebuah pesan memberi tahu Anda bahwa unduhan sedang berlangsung. Setelah unduhan selesai, Anda dapat mengakses `AWS_DMS_Collector_Installer_version_number.msi` file tersebut.

Anda sekarang dapat menginstal pengumpul data DMS pada klien Anda. Untuk informasi selengkapnya, lihat [Menginstal dan mengkonfigurasi pengumpul data](#).

Menginstal dan mengkonfigurasi pengumpul data

Pelajari cara menginstal pengumpul data DMS Anda, cara menentukan kredensial penerusan data, dan cara menambahkan server LDAP ke proyek Anda.

Tabel berikut menjelaskan persyaratan perangkat keras dan perangkat lunak untuk menginstal pengumpul data DMS.

Minimum	Disarankan
Prosesor: 2 core dengan skor benchmark CPU lebih besar dari 8.000	Prosesor: 4 core dengan skor benchmark CPU lebih besar dari 11.000
RAM: 8 GB	RAM: 16 GB
Ukuran hard drive: 256 GB	Ukuran hard drive: 512 GB
Sistem operasi: Microsoft Windows Server 2012 atau lebih tinggi	Sistem operasi: Windows Server 2016 atau lebih tinggi

Untuk menginstal pengumpul data pada klien di jaringan Anda

1. Jalankan installer.MSI. Halaman Wisaya Pengaturan Kolektor Penasihat AWS DMS Armada muncul.
2. Pilih Berikutnya. Perjanjian lisensi pengguna akhir muncul.
3. Baca dan terima perjanjian lisensi Pengguna Akhir.
4. Pilih Berikutnya. Halaman folder Tujuan muncul.
5. Pilih Berikutnya untuk menginstal pengumpul data di direktori default.

Atau pilih Ubah untuk masuk ke direktori instal lain. Lalu pilih Selanjutnya.

6. Pada halaman pintasan Desktop, pilih kotak untuk memasang ikon di desktop Anda.
7. Pilih Instal. Pengumpul data dipasang di direktori yang Anda pilih.
8. Pada halaman Completed DMS Collector Setup Wizard, pilih Launch AWS DMS Collector, dan kemudian pilih Finish.

Pengumpul data DMS Anda menggunakan pustaka, konektor, dan penyedia data .NET untuk terhubung ke database sumber Anda. Penginstal pengumpul data DMS secara otomatis menginstal perangkat lunak yang diperlukan ini untuk semua basis data yang didukung di server Anda.

Setelah Anda menginstal pengumpul data, Anda dapat menjalankannya dari browser dengan memasukkan **http://localhost:11000/** sebagai alamat. Secara opsional, dari menu Start Microsoft Windows, pilih AWS DMSCollector dalam daftar program. Saat pertama kali menjalankan pengumpul data DMS, Anda diminta untuk mengonfigurasi kredensial. Buat nama pengguna dan kata sandi untuk masuk ke pengumpul data.

Di halaman beranda pengumpul data DMS, Anda dapat menemukan informasi untuk menyiapkan dan menjalankan pengumpulan metadata, termasuk kondisi status berikut:

- Status dan kesehatan pengumpulan data Anda.
- Aksesibilitas ke bucket Amazon S3 Anda dan AWS DMS agar pengumpul data dapat meneruskan data ke. AWS DMS
- Konektivitas ke driver database yang diinstal.
- Kredensi server LDAP untuk melakukan penemuan awal.

The screenshot shows the AWS DMS Collector interface. At the top, there's a navigation bar with the AWS logo and 'DMS Collector' text, and a 'Sign out' link. Below the navigation bar is a sidebar with a home icon and a list of icons. The main content area is titled 'Home' and contains three panels:

- Data collection:** Status: Running, Health: 100%.
- Data forwarding:** Name: new-data-collector, Access to Amazon S3: Yes, Access to AWS DMS: Yes, Last uploaded: 31-01-2023 11:43:30. A 'Configure forwarding' button is present.
- Software check (4/4):** Microsoft SQL Server connector for .NET: Passed, MySQL connector for .NET: Passed, Oracle data provider for .NET: Passed, PostgreSQL connector for .NET: Passed.

Below these panels is the 'LDAP servers configuration' section, which includes a '+ Server' button and a table of configured servers:

LDAP server host name ↓	User name	Connection status
<input type="checkbox"/> dc01.dbm.local	shareduser	Passed

Pengumpul data DMS menggunakan direktori LDAP untuk mengumpulkan informasi tentang mesin dan server database di jaringan Anda. Lightweight Directory Access Protocol (LDAP) adalah protokol aplikasi standar terbuka. Ini digunakan untuk mengakses dan memelihara layanan informasi direktori terdistribusi melalui jaringan IP. Anda dapat menambahkan server LDAP yang ada ke proyek Anda untuk pengumpul data yang dapat Anda gunakan untuk menemukan informasi tentang infrastruktur sistem Anda. Untuk melakukannya, pilih opsi +Server, lalu tentukan nama domain yang sepenuhnya memenuhi syarat (FQDN) dan kredensial untuk pengontrol domain Anda. Setelah menambahkan server, validasi cek koneksi. Untuk memulai proses penemuan, lihat [Menemukan server OS dan database untuk dipantau](#).

Mengkonfigurasi kredensi untuk penerusan data

Setelah Anda menginstal pengumpul data, pastikan aplikasi ini dapat mengirim data yang dikumpulkan ke AWS DMS Fleet Advisor.

Untuk mengonfigurasi kredensial untuk penerusan data di Fleet Advisor AWS DMS

1. Pada halaman beranda pengumpul data DMS, di bagian Penerusan data, pilih Konfigurasi penerusan. Kotak dialog Configure credentials for data forwarding terbuka.
2. Pilih Wilayah AWS tempat Anda ingin menggunakan DMS Fleet Advisor.
3. Masukkan ID Kunci AWS Akses dan Kunci Akses AWS Rahasia yang diperoleh sebelumnya saat Anda membuat sumber daya IAM. Untuk informasi selengkapnya, lihat [Buat sumber daya IAM](#).
4. Pilih Jelajahi pengumpul data.

Jika Anda belum membuat pengumpul data di Wilayah yang ditentukan, buat pengumpul data sebelum melanjutkan. Untuk informasi selengkapnya, lihat [Membuat pengumpul data](#).

5. Di jendela Pilih pengumpul data, pilih pengumpul data dalam daftar dan pilih Pilih.
6. Dalam kotak dialog Configure credentials for data forwarding, pilih Simpan.

Pada halaman beranda DMS Collector, di kartu penerusan data, verifikasi bahwa status Access to Amazon S3 dan Access to diatur ke Ya. AWS DMS

Jika Anda melihat bahwa status Akses ke Amazon S3 atau Akses ke diatur ke AWS DMS Tidak, pastikan Anda membuat sumber daya IAM untuk mengakses Amazon S3 dan DMS Fleet Advisor. Setelah Anda membuat sumber daya IAM ini dengan semua izin yang diperlukan, konfigurasi penerusan data lagi. Untuk informasi selengkapnya, lihat [Buat sumber daya IAM](#).

Menemukan server OS dan database untuk dipantau

Anda dapat menggunakan pengumpul data DMS untuk menemukan dan mencantumkan semua server yang tersedia di jaringan Anda. Disarankan untuk menemukan semua server database yang tersedia di jaringan Anda, tetapi tidak diperlukan. Secara opsional, Anda dapat menambahkan atau mengunggah daftar server secara manual untuk pengumpulan data lebih lanjut. Untuk informasi selengkapnya tentang menambahkan daftar server secara manual, lihat [Mengelola objek yang dipantau](#).

Kami menyarankan Anda menemukan semua server sistem operasi (OS) sebelum menemukan database di server tersebut. Untuk menemukan server OS, Anda memerlukan izin untuk menjalankan skrip dan perintah jarak jauh PowerShell, Secure Shell (SSH), dan Windows Management Instrumentation (WMI), serta akses ke registri Windows. Untuk menemukan server database di jaringan Anda dan mengumpulkan metadata dari mereka, Anda memerlukan izin administrator read-only untuk koneksi database jarak jauh. Pastikan Anda menambahkan server LDAP sebelum melanjutkan dengan penemuan. Untuk informasi selengkapnya, lihat [Mengkonfigurasi kredensi untuk penerusan data](#).

Untuk memulai dengan pengumpul data DMS, selesaikan tugas-tugas berikut:

- Temukan semua server OS di jaringan Anda.
- Tambahkan server OS tertentu sebagai objek untuk dipantau.
- Verifikasi koneksi untuk server OS yang dipantau.
- Temukan database Microsoft SQL Server, MySQL, Oracle, dan PostgreSQL yang berjalan di server OS.
- Tambahkan server database untuk pengumpulan data.
- Verifikasi koneksi ke database yang dipantau.

Untuk menemukan server OS di jaringan Anda yang dapat Anda pantau

1. Di panel navigasi pengumpul data DMS, pilih Discovery. Untuk menampilkan panel navigasi, pilih ikon menu di sudut kiri atas halaman beranda pengumpul data DMS.

Halaman Discovery terbuka.

2. Pastikan tab server OS dipilih, lalu pilih Jalankan penemuan. Kotak dialog parameter Discovery muncul.
3. Masukkan server LDAP yang ingin Anda gunakan untuk memindai jaringan Anda.
4. Pilih Jalankan penemuan. Halaman ini menampilkan daftar semua server OS yang ditemukan dalam jaringan Anda, terlepas dari apakah mereka menjalankan database.

Kami menyarankan Anda menjalankan penemuan untuk semua server OS sebelum menjalankan penemuan untuk database di server tersebut. Kredensial Anda memungkinkan penemuan pertama untuk server host, kemudian untuk database yang berada di dalamnya. Anda ingin menemukan server OS terlebih dahulu sebelum menjalankan penemuan untuk database di server tersebut. Ketahuilah bahwa kredensial yang Anda gunakan untuk server LDAP untuk

menemukan server OS di jaringan Anda mungkin berbeda dari kredensial yang diperlukan untuk menemukan database pada server OS tertentu. Oleh karena itu, kami menyarankan Anda menambahkan server OS ke objek yang dipantau, memverifikasi kredensialnya dan memperbaikinya jika perlu, dan kemudian memeriksa konektivitas sebelum melanjutkan.

Dalam daftar server OS yang ditemukan di jaringan Anda, Anda sekarang dapat memilih server yang ingin Anda tambahkan ke objek yang dipantau.

Untuk memilih server OS sebagai objek untuk dipantau

1. Pada halaman Discovery, pilih tab Server OS.
2. Dalam daftar server OS yang ditemukan yang ditampilkan, pilih kotak centang di samping setiap server yang ingin Anda pantau.
3. Pilih Tambahkan ke objek yang dipantau.

Anda dapat melihat daftar server OS untuk memantau dan memverifikasi koneksi pada halaman Objek Monitor.

Untuk memverifikasi koneksi server OS yang dipilih untuk dipantau

1. Di panel navigasi pengumpul data DMS, pilih Objek yang dipantau.
2. Pada halaman Objek yang dipantau, pilih tab Server OS. Daftar server OS yang ditemukan untuk dipantau muncul.
3. Pilih kotak centang di bagian atas kolom untuk memilih semua server OS yang terdaftar.
4. Pilih Tindakan, lalu Verifikasi koneksi. Untuk setiap objek server, lihat hasilnya di kolom status Koneksi.
5. Pilih server dengan status koneksi selain Sukses. Selanjutnya, pilih Tindakan, lalu pilih Edit. Kotak dialog Edit server terbuka.
6. Verifikasi bahwa informasi tersebut benar atau edit jika diperlukan. Setelah selesai, silakan pilih Simpan. Kotak dialog Override credentials terbuka.
7. Pilih Timpa. Pengumpul data DMS memverifikasi dan memperbarui status untuk setiap koneksi sebagai Sukses.

Anda sekarang dapat menemukan database yang berada di server yang Anda pilih untuk dipantau.

Temukan database yang berjalan di server

1. Di panel navigasi pengumpul data DMS, pilih Discovery.
2. Pilih tab Server database, dan pilih Jalankan penemuan. Kotak dialog parameter Discovery terbuka.
3. Di kotak dialog Parameter Discovery, untuk Discovery by, pilih Objek yang dipantau. Untuk Server, pilih server OS tempat Anda ingin menjalankan penemuan basis data.
4. Pilih Jalankan penemuan. Halaman ini menampilkan daftar semua database yang berada di server OS yang Anda pilih untuk dipantau.

Lihat informasi seperti alamat database, nama server, dan mesin database untuk membantu Anda memilih database untuk dipantau.

Untuk memilih database untuk dipantau

1. Pada halaman Discovery, pilih tab Server database.
2. Dalam daftar database yang ditemukan yang ditampilkan, pilih kotak centang di sebelah semua database yang ingin Anda pantau.
3. Pilih Tambahkan ke objek yang dipantau.

Anda sekarang dapat memverifikasi koneksi ke database yang Anda pilih untuk dipantau.

Untuk memverifikasi koneksi ke database yang dipantau

1. Di panel navigasi pengumpul data DMS, pilih Objek yang dipantau.
2. Pada halaman Objek yang dipantau, pilih tab Server database. Daftar server database yang ditemukan yang Anda pilih untuk dipantau muncul.
3. Pilih kotak centang di bagian atas kolom untuk memilih semua server database yang terdaftar.
4. Pilih Tindakan, lalu pilih Verifikasi koneksi. Untuk setiap database, lihat hasilnya di kolom Status koneksi.
5. Pilih koneksi yang memiliki status tidak terdefinisi (kosong) atau status Kegagalan. Selanjutnya, pilih Tindakan, lalu pilih Edit. Kotak dialog Edit objek yang dipantau terbuka.
6. Masukkan kredensi Login dan Kata Sandi Anda, lalu pilih Simpan. Kotak dialog Change credentials terbuka.

7. Pilih Timpa. Pengumpul data DMS memverifikasi dan memperbarui status untuk setiap koneksi sebagai Sukses.

Setelah menemukan server OS dan database untuk dipantau, Anda juga dapat melakukan tindakan untuk mengelola objek yang dipantau.

Mengelola objek yang dipantau

Anda dapat memilih objek untuk dipantau ketika Anda menjalankan proses penemuan server seperti yang dijelaskan dalam [Menemukan server OS dan database](#). Selain itu, Anda dapat mengelola objek secara manual, seperti server sistem operasi (OS) dan server database. Anda dapat melakukan tindakan berikut untuk mengelola objek yang dipantau:

- Tambahkan objek baru untuk dipantau
- Hapus objek yang ada
- Edit objek yang ada
- Ekspor dan impor daftar objek untuk dipantau
- Periksa koneksi ke objek
- Mulai pengumpulan data

Misalnya, Anda dapat menambahkan objek secara manual ke monitor.

Untuk menambahkan objek untuk dipantau secara manual

1. Pada halaman Objek yang Dipantau, pilih +Server. Kotak dialog Add monitored object terbuka.
2. Tambahkan informasi tentang server, lalu pilih Simpan.

Anda juga dapat menggunakan .csv file untuk mengimpor daftar besar objek untuk dipantau. Gunakan format .csv file berikut untuk mengimpor daftar objek ke pengumpul data 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
```

Untuk mengimpor file.csv dengan daftar objek untuk dipantau

1. Pilih Impor. Halaman Impor objek yang dipantau terbuka.
2. Jelajahi .csv file yang ingin Anda impor, lalu pilih Berikutnya.

Anda dapat melihat semua objek dan memilih objek yang ingin Anda mulai mengumpulkan metadata.

Mengaitkan server OS dengan database yang ditambahkan secara manual

DMS Fleet Advisor tidak dapat mengumpulkan metrik kinerja langsung dari database MySQL dan PostgreSQL. Untuk mengumpulkan metrik yang diperlukan untuk rekomendasi target, DMS Fleet Advisor menggunakan metrik OS tempat database Anda berjalan.

Saat Anda menambahkan database MySQL dan PostgreSQL secara manual ke daftar objek yang dipantau, pengumpul data DMS tidak dapat mengidentifikasi server OS tempat database ini berjalan. Karena masalah ini, Anda harus mengaitkan database MySQL dan PostgreSQL Anda dengan server OS.

Anda tidak perlu mengaitkan server OS secara manual dengan database yang secara otomatis ditemukan oleh DMS Fleet Advisor.

Untuk mengaitkan server OS dengan database Anda

1. Di panel navigasi pengumpul data DMS, pilih Objek yang dipantau.
2. Pada halaman Objek yang dipantau, pilih tab Server database. Daftar server database muncul.
3. Pilih kotak centang di samping server database MySQL atau PostgreSQL yang Anda tambahkan secara manual.

4. Pilih Tindakan, lalu pilih Edit. Kotak dialog Edit database terbuka.
5. Jika pengumpul data DMS Anda telah menemukan server OS tempat database ini berjalan, maka pilih Deteksi otomatis. Pengumpul data DMS menjalankan skrip SQL untuk secara otomatis mengidentifikasi server OS tempat database Anda berjalan. Kemudian, pengumpul data DMS mengaitkan server OS ini dengan database Anda. Lewati langkah berikutnya dan simpan konfigurasi database yang Anda edit.

Jika pengumpul data DMS tidak dapat secara otomatis mengidentifikasi server OS untuk database Anda, pastikan Anda menggunakan kredensyal yang benar dan memberikan izin akses database. Secara opsional, Anda dapat menambahkan server OS secara manual.

6. Untuk menambahkan server OS Anda secara manual, pilih+Tambahkan server OS. Kotak dialog Add host OS server terbuka.

Tambahkan informasi tentang server OS Anda, lalu pilih Simpan.

7. Di kotak dialog Edit database, pilih Verifikasi koneksi untuk memastikan bahwa pengumpul data DMS Anda dapat terhubung ke server OS.
8. Setelah Anda memverifikasi koneksi, pilih Simpan.

Jika Anda mengubah server OS terkait untuk basis data sumber Anda, maka DMS Fleet Advisor menggunakan metrik yang diperbarui untuk menghasilkan rekomendasi. Namun, CloudWatch bagan Amazon menampilkan data lama untuk server database Anda. Untuk informasi selengkapnya tentang CloudWatch grafik, lihat [Rincian rekomendasi](#).

Menggunakan SSL dengan AWS DMS Fleet Advisor

Untuk melindungi data Anda, AWS DMS Fleet Advisor dapat menggunakan SSL untuk mengakses database Anda.

Database yang didukung

AWS DMS Fleet Advisor mendukung penggunaan SSL untuk mengakses database berikut:

- Microsoft SQL Server
- MySQL
- PostgreSQL

Menyiapkan SSL

Untuk menggunakan SSL untuk mengakses database Anda, konfigurasi server database Anda untuk mendukung SSL. Untuk informasi selengkapnya, lihat dokumentasi berikut untuk database Anda:

- SQL Server: [Aktifkan koneksi terenkripsi ke Mesin Database](#)
- MySQL: [Mengkonfigurasi MySQL untuk Menggunakan Koneksi Terenkripsi](#)
- PostgreSQL: [Koneksi TCP/IP Aman dengan SSL](#)

Untuk menggunakan SSL untuk terhubung ke database Anda, pilih Percayai sertifikat server dan Gunakan SSL saat menambahkan server secara manual. Untuk database MySQL, Anda dapat menggunakan sertifikat kustom. Untuk menggunakan sertifikat kustom, pilih kotak centang Verifikasi CA. Untuk informasi tentang menambahkan server, lihat [Mengelola objek yang dipantau](#).

Memeriksa Sertifikat Otoritas Sertifikat Server (CA) untuk SQL Server

Jika Anda ingin memvalidasi Sertifikat Otoritas Sertifikat Server (CA) untuk SQL Server, maka hapus sertifikat server Trust saat Anda menambahkan server. Jika server Anda menggunakan CA yang terkenal, dan CA diinstal secara default pada OS Anda, maka verifikasi harus bekerja secara normal. Jika DMS Fleet Advisor tidak dapat terhubung ke server database Anda, instal sertifikat CA yang digunakan server database Anda. Untuk informasi selengkapnya, lihat [Mengkonfigurasi klien](#).

Mengumpulkan data untuk AWS DMS Fleet Advisor

Untuk mulai mengumpulkan data, pilih objek pada halaman Objek yang dipantau, dan pilih Jalankan pengumpulan data. Pengumpul data DMS dapat mengumpulkan hingga 100 database sekaligus. Selain itu, pengumpul data DMS dapat menggunakan hingga delapan thread paralel untuk terhubung ke database di lingkungan Anda. Dari delapan thread ini, pengumpul data DMS dapat menggunakan hingga lima thread paralel untuk terhubung ke satu instance database.

Important

Sebelum mulai mengumpulkan data, lihat bagian pemeriksaan Perangkat Lunak di halaman beranda pengumpul data DMS. Verifikasi bahwa semua mesin database yang ingin Anda pantau memiliki status Lulus. Jika beberapa mesin database memiliki status Gagal, dan Anda memiliki server database dengan mesin yang sesuai dalam daftar objek yang dipantau,

perbaiki masalah sebelum melanjutkan. Anda dapat menemukan tips di samping status Gagal yang tercantum di bagian Pemeriksaan perangkat lunak.

Pengumpul data DMS dapat bekerja dalam dua mode: single run atau monitoring berkelanjutan. Setelah Anda memulai pengumpulan data, kotak dialog Jalankan pengumpulan data terbuka. Selanjutnya, pilih salah satu dari dua opsi berikut.

Kapasitas metadata dan database

Pengumpul data DMS mengumpulkan informasi dari database atau server OS. Ini termasuk skema, versi, edisi, CPU, memori, dan kapasitas disk. Pengumpul data DMS juga mengumpulkan dan menyediakan metrik seperti IOPS, throughput I/O dan koneksi server basis data aktif. Anda dapat menghitung rekomendasi target di DMS Fleet Advisor berdasarkan informasi ini. Jika basis data sumber terlalu banyak atau kurang, maka rekomendasi target juga akan lebih-atau kurang.

Ini adalah pilihan default.

Metadata, kapasitas database, dan pemanfaatan sumber daya

Selain metadata dan informasi kapasitas database, pengumpul data DMS mengumpulkan metrik pemanfaatan aktual CPU, memori, dan kapasitas disk untuk database atau server OS. Pengumpul data DMS juga mengumpulkan dan menyediakan metrik seperti IOPS, throughput I/O dan koneksi server basis data aktif. Rekomendasi target yang diberikan akan lebih akurat karena didasarkan pada beban kerja database yang sebenarnya.

Jika Anda memilih opsi ini, maka Anda mengatur periode pengumpulan data. Anda dapat mengumpulkan data selama 7 hari berikutnya atau mengatur rentang Kustom 1-60 hari.

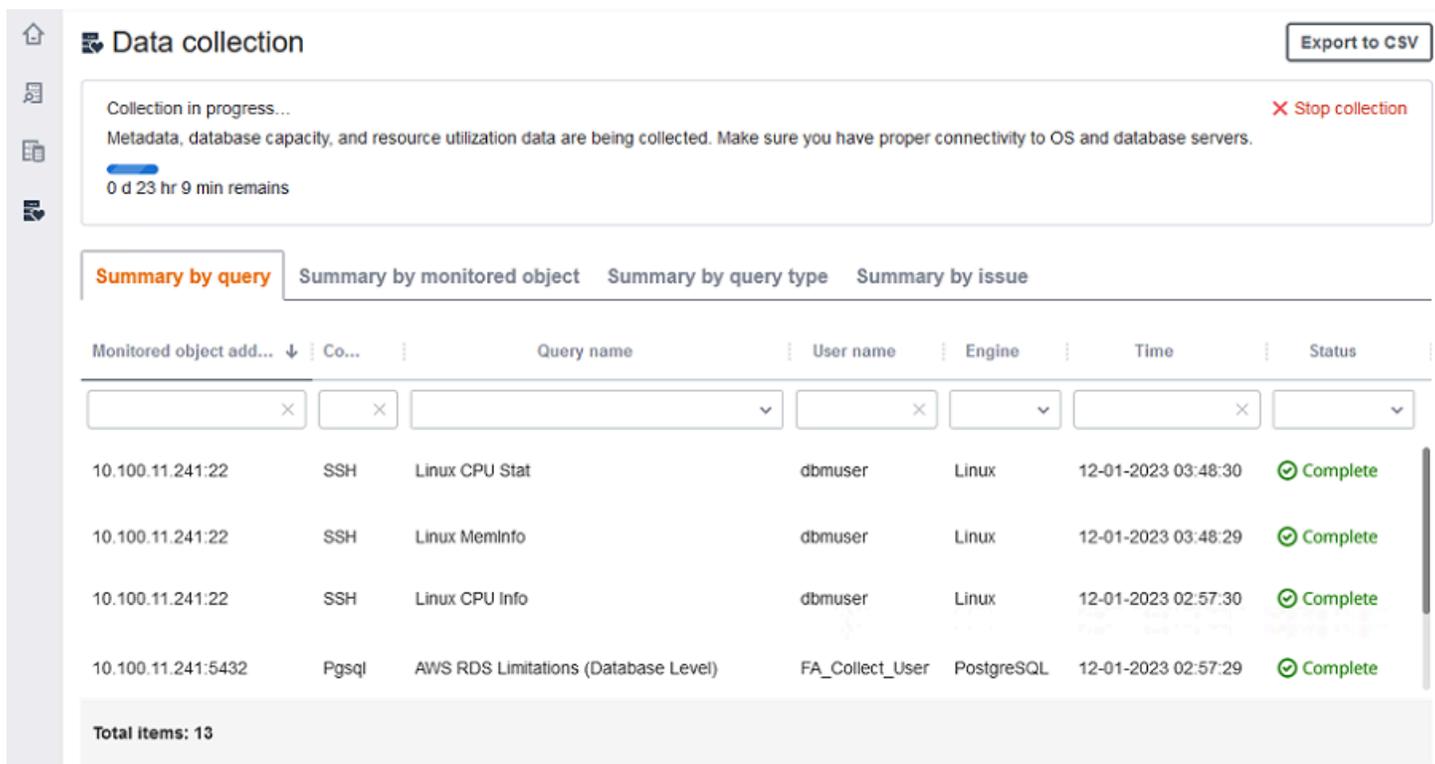
Setelah pengumpulan data dimulai, Anda akan diarahkan ke halaman pengumpulan data, di mana Anda dapat melihat bagaimana kueri pengumpulan berjalan dan memantau kemajuan langsung. Di sini, Anda dapat melihat kesehatan koleksi secara keseluruhan atau di halaman beranda pengumpul data DMS. Jika kesehatan pengumpulan data secara keseluruhan kurang dari 100 persen, Anda mungkin perlu memperbaiki masalah yang terkait dengan pengumpulan.

Jika Anda menjalankan pengumpul data DMS dalam mode Metadata dan kapasitas database, maka Anda dapat melihat jumlah kueri yang diselesaikan pada halaman Pengumpulan data.

Jika Anda menjalankan pengumpul data DMS dalam Metadata, kapasitas database, dan mode pemanfaatan sumber daya, maka Anda dapat melihat sisa waktu sebelum pengumpul data DMS Anda menyelesaikan pemantauan.

Pada halaman Pengumpulan data, Anda dapat melihat status pengumpulan untuk setiap objek. Jika sesuatu tidak berfungsi dengan baik, muncul pesan yang menunjukkan berapa banyak masalah yang terjadi. Untuk membantu menentukan perbaikan masalah, Anda dapat memeriksa detailnya. Tab berikut mencantumkan potensi masalah:

- Ringkasan berdasarkan kueri - Menunjukkan status tes seperti tes Ping. Anda dapat memfilter hasil di kolom Status. Kolom Status menyediakan pesan yang menunjukkan berapa banyak kegagalan yang terjadi selama pengumpulan data.
- Ringkasan oleh objek yang dipantau - Menunjukkan status keseluruhan per objek.
- Ringkasan berdasarkan jenis kueri - Menampilkan status untuk jenis kueri kolektor, seperti panggilan SQL, Secure Shell (SSH), atau Windows Management Instrumentation (WMI).
- Ringkasan berdasarkan masalah - Menampilkan semua masalah unik yang terjadi, dengan nama masalah dan berapa kali setiap masalah terjadi.



The screenshot shows the 'Data collection' page in the AWS DMS console. At the top, there is a status box indicating 'Collection in progress...' with a progress bar and a 'Stop collection' button. Below this, there are four tabs for summarizing the data: 'Summary by query' (selected), 'Summary by monitored object', 'Summary by query type', and 'Summary by issue'. The main content is a table with the following columns: 'Monitored object address', 'Collection type', 'Query name', 'User name', 'Engine', 'Time', and 'Status'. The table lists four completed queries, all with a status of 'Complete'.

Monitored object address	Collection type	Query name	User name	Engine	Time	Status
10.100.11.241:22	SSH	Linux CPU Stat	dbmuser	Linux	12-01-2023 03:48:30	Complete
10.100.11.241:22	SSH	Linux MemInfo	dbmuser	Linux	12-01-2023 03:48:29	Complete
10.100.11.241:22	SSH	Linux CPU Info	dbmuser	Linux	12-01-2023 02:57:30	Complete
10.100.11.241:5432	Pgsql	AWS RDS Limitations (Database Level)	FA_Collect_User	PostgreSQL	12-01-2023 02:57:29	Complete

Total items: 13

Untuk mengeksport hasil koleksi, pilih Ekspor ke CSV.

Setelah mengidentifikasi masalah dan menyelesaikannya, pilih Mulai pengumpulan dan jalankan kembali proses pengumpulan data. Setelah melakukan pengumpulan data, pengumpul data menggunakan koneksi aman untuk mengunggah data yang dikumpulkan ke inventaris DMS Fleet Advisor. DMS Fleet Advisor menyimpan informasi di bucket Amazon S3 Anda. Untuk informasi tentang mengonfigurasi kredensial untuk penerusan data, lihat. [Mengkonfigurasi kredensi untuk penerusan data](#)

Mengumpulkan metrik kapasitas dan pemanfaatan sumber daya dengan AWS DMS Fleet Advisor

Anda dapat mengumpulkan metadata dan metrik kinerja dalam dua mode: pemantauan tunggal atau pemantauan berkelanjutan. Bergantung pada opsi yang Anda pilih, pengumpul data DMS Anda melacak metrik yang berbeda di lingkungan data Anda. Selama satu kali proses, pengumpul data DMS Anda hanya melacak metadata metrik dari database dan server OS Anda. Selama pemantauan berkelanjutan, pengumpul data DMS Anda melacak pemanfaatan sumber daya Anda yang sebenarnya.

AWS DMS mengumpulkan metadata dan metrik berikut selama satu kali menjalankan pengumpul data DMS Anda.

- Memori yang tersedia di server OS Anda
- Penyimpanan yang tersedia di server OS Anda
- Versi dan edisi basis data
- Jumlah CPU di server OS Anda
- Jumlah skema
- Jumlah prosedur yang disimpan
- Jumlah tabel
- Jumlah pemicu
- Jumlah tampilan
- Struktur skema

DMS Fleet Advisor menggunakan metrik ini untuk membangun inventaris database dan server OS Anda. Selain itu, DMS Fleet Advisor menggunakan metadata dan metrik ini untuk menganalisis skema basis data sumber Anda.

DMS Fleet Advisor dapat menghasilkan rekomendasi target menggunakan metrik yang dikumpulkan selama satu kali pengumpulan data. Namun, dalam hal ini untuk basis data sumber Anda yang terlalu banyak, rekomendasi target juga dilebih-lebihkan. Dengan demikian, Anda dikenakan biaya tambahan untuk pemeliharaan sumber daya Anda di AWS Cloud Untuk database sumber underprovisioned, rekomendasi target juga underprovisioned, yang dapat menyebabkan masalah kinerja. Kami merekomendasikan untuk mengumpulkan data menggunakan pemantauan berkelanjutan dengan memilih metadata, kapasitas database, dan mode pemanfaatan sumber daya untuk pengumpul data DMS.

AWS DMS mengumpulkan metrik berikut selama pemantauan yang sedang berlangsung. Anda dapat menjalankan pengumpul data DMS Anda untuk jangka waktu 1 hingga 60 hari.

- I/O throughput pada server database Anda
- Operasi input/output per detik (IOPS) di server database Anda
- Jumlah CPU yang digunakan server OS Anda
- Penggunaan memori di server OS Anda
- Jumlah database aktif dan koneksi server OS

DMS Fleet Advisor menggunakan metrik ini untuk menghasilkan rekomendasi target yang akurat, sehingga basis data target Anda memenuhi kebutuhan kinerja Anda. Ini dapat mencegah biaya tambahan yang dikeluarkan untuk pemeliharaan sumber daya Anda di AWS Cloud

Bagaimana Penasihat AWS DMS Armada mengumpulkan metrik kapasitas dan pemanfaatan sumber daya?

DMS Fleet Advisor mengumpulkan metrik kinerja setiap menit.

Untuk Oracle dan SQL Server, DMS Fleet Advisor menjalankan kueri SQL untuk menangkap nilai untuk setiap metrik database.

Untuk MySQL dan PostgreSQL, DMS Fleet Advisor mengumpulkan metrik kinerja dari server OS tempat database Anda berjalan. Di Windows, DMS Fleet Advisor menjalankan skrip WMI Query Language (WQL) dan menerima data WMI. Di Linux, DMS Fleet Advisor menjalankan perintah yang menangkap metrik server OS.

⚠ Important

Menjalankan skrip SQL jarak jauh dapat memengaruhi kinerja basis data produksi Anda. Namun, kueri pengumpulan data tidak mengandung logika perhitungan apa pun. Dengan demikian, proses pengumpulan data tidak mungkin menggunakan lebih dari 1 persen sumber daya database Anda.

Anda dapat melihat semua kueri yang dijalankan pengumpul data untuk mengumpulkan metrik. Untuk melakukannya, buka `DMSCollector.Collections.json` file. Anda dapat menemukan file ini di `etc` folder yang terletak di folder yang sama tempat Anda menginstal pengumpul data. Jalur defaultnya adalah `C:\ProgramData\Amazon\AWS DMS Collector\etc\DMSCollector.Collections.json`.

Pengumpul data DMS menggunakan sistem file lokal sebagai penyimpanan sementara untuk semua data yang dikumpulkan. Pengumpul data DMS menyimpan data yang dikumpulkan dalam format JSON. Anda dapat menggunakan kolektor lokal dalam mode offline dan secara manual memeriksa atau memverifikasi file yang dikumpulkan sebelum Anda mengonfigurasi penerusan data. Anda dapat melihat semua file yang dikumpulkan di `out` folder yang terletak di folder yang sama tempat Anda menginstal pengumpul data DMS. Jalur defaultnya adalah `C:\ProgramData\Amazon\AWS DMS Collector\out`.

⚠ Important

Jika Anda menjalankan pengumpul data DMS dalam mode offline dan menyimpan data yang dikumpulkan di server Anda selama lebih dari 14 hari, maka Anda tidak dapat menggunakan Amazon CloudWatch untuk menampilkan metrik ini. Namun, DMS Fleet Advisor masih menggunakan data ini untuk menghasilkan rekomendasi. Untuk informasi selengkapnya tentang CloudWatch grafik, lihat [Rincian rekomendasi](#).

Anda juga dapat memeriksa atau memverifikasi file data yang dikumpulkan dalam mode online. Pengumpul data DMS meneruskan semua data ke bucket Amazon S3 yang Anda tentukan dalam pengaturan pengumpul data DMS.

Anda dapat menggunakan pengumpul data DMS untuk mengumpulkan data dari database lokal. Selain itu, Anda dapat mengumpulkan data dari database Amazon RDS dan Aurora. Namun, Anda tidak dapat berhasil menjalankan semua kueri pengumpul data DMS di cloud karena perbedaan

antara Amazon RDS atau Aurora dan instans DB lokal. Karena pengumpul data DMS mengumpulkan metrik pemanfaatan untuk database MySQL dan PostgreSQL dari OS host, pendekatan ini tidak akan bekerja dengan Amazon RDS dan Aurora.

Pemecahan masalah untuk pengumpul data DMS

Dalam daftar berikut, Anda dapat menemukan tindakan yang harus diambil ketika Anda mengalami masalah tertentu saat mengumpulkan data dengan pengumpul data Anda.

Topik

- [Masalah pengumpulan data yang terkait dengan koneksi jaringan dan server](#)
- [Masalah pengumpulan data yang terkait dengan Instrumentasi Manajemen Windows](#)
- [Masalah pengumpulan data yang terkait dengan komposer halaman web Windows](#)
- [Masalah pengumpulan data yang terkait dengan SSL](#)

Masalah pengumpulan data yang terkait dengan koneksi jaringan dan server

NET: Pengecualian terjadi selama permintaan ping.

Periksa nama komputer untuk melihat apakah itu dalam keadaan di mana tidak dapat diselesaikan ke alamat IP.

Misalnya, periksa apakah komputer dimatikan, terputus dari jaringan, atau dinonaktifkan.

NET: Habis Waktu

Aktifkan aturan firewall masuk “Berbagi File dan Printer (Permintaan Gema - ICMPv4-in)”.
Sebagai contoh:

* Inbound ICMPv4

BERSIH: DestinationHostUnreachable

Periksa alamat IP komputer. Secara khusus, periksa apakah itu pada subnet yang sama dengan komputer yang menjalankan pengumpul data DMS dan apakah itu menanggapi permintaan Address Resolution Protocol (ARP).

Jika komputer berada di subnet yang berbeda, maka alamat IP gateway tidak dapat diselesaikan ke alamat kontrol akses media (MAC).

Juga, periksa apakah komputer dimatikan, terputus dari jaringan, atau dinonaktifkan.

Masalah pengumpulan data yang terkait dengan Instrumentasi Manajemen Windows

WMI: Server RPC tidak tersedia. (Pengecualian dari HRESULT: 0x800706BA)

Aktifkan aturan firewall masuk “Windows Management Instrumentation (DCOM — in)”. Sebagai contoh:

- * Inbound TCP/IP at local port 135.

Juga, aktifkan aturan firewall masuk “Windows Management Instrumentation (WMI-in)”. Sebagai contoh:

- * Inbound TCP/IP at local port 49152 - 65535 untuk Windows Server 2008 dan versi yang lebih tinggi.

- * Inbound TCP/IP at local port 1025 - 5000 untuk Windows Server 2003 dan versi yang lebih rendah.

WMI: Akses ditolak. (Pengecualian dari HRESULT: 0x80070005 (E_ACCESSDENIED))

Coba yang berikut ini:

- Tambahkan pengguna pengumpul data DMS ke grup Windows, Pengguna COM Terdistribusi atau Administrator.
- Mulai layanan Instrumentasi Manajemen Windows dan atur tipe start-upnya ke Otomatis.
- Pastikan nama pengguna pengumpul data DMS Anda ada dalam \ format.

WMI: Akses ditolak

Tambahkan izin Aktifkan Jarak Jauh ke pengguna pengumpul data DMS di namespace WMI root.

Gunakan pengaturan lanjutan dan pastikan bahwa izin berlaku untuk “Namespace dan subnamespace ini.”

WMI: Panggilan dibatalkan oleh filter pesan. (Pengecualian dari HRESULT: 0x80010002...)

Mulai ulang layanan Instrumentasi Manajemen Windows.

Masalah pengumpulan data yang terkait dengan komposer halaman web Windows

WPC: Jalur jaringan tidak ditemukan

Aktifkan aturan firewall masuk “Berbagi File dan Printer (SMB—in)”. Sebagai contoh:

* Inbound TCP/IP at local port 445.

Juga, mulai layanan Remote Registry dan atur tipe start-upnya ke Otomatis.

WPC: Akses ditolak

Tambahkan pengguna pengumpul data DMS ke grup Pengguna Monitor Kinerja atau Administrator.

WPC: Kategori tidak ada

Jalankan `loader /r` untuk membangun kembali cache penghitung kinerja, lalu restart komputer Anda.

Note

Untuk informasi tentang pemecahan masalah saat memigrasi data menggunakan AWS Database Migration Service (AWS DMS), lihat [Pemecahan masalah](#) dan dukungan diagnostik.

Masalah pengumpulan data yang terkait dengan SSL

Kesalahan SSL

Database Anda memerlukan koneksi SSL yang aman, dan Anda belum mengaktifkan opsi Verifikasi CA dan Gunakan SSL untuk koneksi. Aktifkan opsi ini dan pastikan bahwa OS lokal Anda memiliki Otoritas Sertifikat yang diinstal yang digunakan database Anda. Lihat informasi yang lebih lengkap di [Menyiapkan SSL](#).

Menggunakan inventaris untuk analisis di AWS DMS Fleet Advisor

Untuk memeriksa kelayakan migrasi database potensial, Anda dapat bekerja dengan inventaris database dan skema yang ditemukan. Anda dapat menggunakan informasi dalam inventaris ini untuk memahami database dan skema mana yang merupakan kandidat yang baik untuk migrasi.

Anda dapat mengakses inventaris database dan skema di konsol. Untuk melakukannya, pilih Inventaris di konsol.

The screenshot shows the AWS DMS console interface. On the left is a navigation menu with options like Dashboard, Discover, Assess, Convert, and Migrate data. The main content area is titled 'Inventory' and includes a sub-header 'Database servers, schema information, and metadata discovered by data collectors.' A large orange button labeled 'Analyze Inventories' is positioned in the top right. Below this, a blue information box explains the analysis process. The 'Databases' tab is active, showing a table with columns for Database, Server, Number of objects, and Engine. One database entry is visible, identified as a Microsoft SQL Server.

DMS Fleet Advisor menganalisis skema database Anda untuk menentukan kesamaan skema yang berbeda. Analisis ini tidak membandingkan kode sebenarnya untuk objek. DMS Fleet Advisor hanya membandingkan nama-nama objek skema, seperti fungsi dan prosedur, untuk mengidentifikasi objek serupa dalam skema database yang berbeda.

Topik

- [Menggunakan inventaris database untuk analisis](#)
- [Menggunakan inventaris skema untuk analisis](#)

Menggunakan inventaris database untuk analisis

Untuk melihat daftar semua database pada semua server yang ditemukan dalam jaringan Anda dari mana data dikumpulkan, gunakan prosedur berikut.

Untuk melihat daftar database di server jaringan Anda tempat data dikumpulkan

1. Pilih Inventaris di konsol.

Halaman Inventaris terbuka.

2. Pilih tab Database.

Daftar database yang ditemukan muncul.

Inventory Info

Database servers, schema information, and metadata discovered by data collectors. [Analyze inventories](#)

Analyze inventories
Running the analysis helps in identifying the candidates for migration. All the schemas are analyzed when you take this action, so ensure that the inventory is complete before you run the analysis. This operation can take a few minutes. [Learn more](#)

Databases | Schemas

Databases (7) [Refresh](#) [Export to CSV](#) [Delete](#)

Database inventories that were discovered by data collectors.

Find database inventory

<input type="checkbox"/>	Database	Server	Number of s...	Engine	Engine version	Engine ...
<input type="checkbox"/>	WinServ2016.d...	-	No data	PostgreSQL	-	-
<input type="checkbox"/>	VM-MSSQL14-...	10.11.1.10	44	Microsoft SQL ...	2014 (Extended support)	Enterprise
<input type="checkbox"/>	MSSQL01.dbm...	-	No data	Microsoft SQL ...	2019 (Mainstream support)	Express

- Pilih Analisis inventaris untuk menentukan properti skema, seperti kesamaan dan kompleksitas. Jumlah waktu yang dibutuhkan proses tergantung pada jumlah objek yang akan dianalisis, tetapi tidak akan memakan waktu lebih dari satu jam. Hasil dari analisis ditemukan di tab Skema yang terletak di halaman Inventaris.

DMS Fleet Advisor menganalisis skema di semua database yang ditemukan untuk menentukan persimpangan objek mereka. Hasil analisis dinyatakan dalam persentase. DMS Fleet Advisor menganggap skema dengan persimpangan lebih dari 50 persen sebagai duplikat. Skema asli diidentifikasi sebagai skema yang ditemukan duplikat. Ini membantu mengidentifikasi skema asli untuk dikonversi atau dimigrasi terlebih dahulu.

Seluruh inventaris dianalisis bersama untuk mengidentifikasi skema duplikat.

Menggunakan inventaris skema untuk analisis

Anda dapat melihat daftar skema database yang ditemukan di server dalam jaringan Anda dari mana data dikumpulkan. Lakukan prosedur berikut.

Untuk melihat daftar skema di server jaringan tempat data dikumpulkan

- Pilih Inventaris di konsol. Halaman Inventaris terbuka.

2. Pilih tab Skema. Daftar skema muncul.
3. Pilih skema dalam daftar untuk melihat informasi tentangnya, termasuk server, database, ukuran, dan kompleksitas.

Untuk setiap skema, Anda dapat melihat ringkasan objek yang menyediakan informasi tentang jenis objek, jumlah objek, ukuran objek, dan baris kode.

4. (Opsional) Pilih Analisis inventaris untuk mengidentifikasi skema duplikat. DMS Fleet Advisor menganalisis skema database untuk menentukan persimpangan objek mereka.
5. Anda dapat mengekspor informasi inventaris ke .csv file untuk ditinjau lebih lanjut.

The screenshot displays the 'Inventory' section of the AWS SCT console. At the top, a green banner indicates 'Analysis complete'. Below this, a navigation breadcrumb shows 'DMS > Discover: Inventory'. The main heading is 'Inventory' with an 'Info' link. A sub-heading reads 'Database servers, schema information, and metadata discovered by data collectors.' To the right, there is an orange button labeled 'Analyze inventories'. A light blue notification box contains an information icon and text: 'Analyze inventories. Running the analysis helps in identifying the candidates for migration. All the schemas are analyzed when you take this action, so ensure that the inventory is complete before you run the analysis. This operation can take a few minutes. Learn more [link]'. Below the notification, there are tabs for 'Databases' and 'Schemas', with 'Schemas' selected. The 'Schemas (13)' section includes a search bar with the placeholder 'Find schema inventory', a refresh button, and an 'Export to CSV' button. A table lists the discovered schemas with the following columns: Schema, Server, Database, Engine, Complexity, Similarity..., and Original schema. The table contains four rows of data.

Schema	Server	Database	Engine	Complexity	Similarity...	Original schema
lsa_tests_src.lsa_tests_src	linuxsql02.db.local	linuxsql02.db.local:3306	MySQL Server	Simple	100	lsa_tests_src_100.lsa_tests_s...
lsa_tests_src_90e_30a.lsa_t...	linuxsql02.db.local	linuxsql02.db.local:3306	MySQL Server	Simple	90	lsa_tests_src_49.lsa_tests_sr..
lsa_tests_src_50.lsa_tests_s...	linuxsql02.db.local	linuxsql02.db.local:3306	MySQL Server	Simple	50	lsa_tests_src_100.lsa_tests_s...
lsa_tests_src_49.lsa_tests_s...	linuxsql02.db.local	linuxsql02.db.local:3306	MySQL Server	Simple	-	None

Untuk mengidentifikasi skema yang akan dimigrasi dan menentukan target migrasi, Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) atau Konversi Skema DMS. Untuk informasi selengkapnya, lihat [Menggunakan wizard proyek baru di AWS SCT](#).

Setelah mengidentifikasi skema untuk dimigrasi, Anda dapat mengonversi skema menggunakan AWS SCT atau Konversi Skema DMS. Untuk informasi selengkapnya tentang Konversi Skema DMS, lihat [Mengonversi skema database menggunakan Konversi Skema DMS](#)

Menggunakan fitur Rekomendasi Target Penasihat AWS DMS Armada

Untuk menjelajahi dan memilih target migrasi yang optimal, Anda dapat membuat rekomendasi target untuk basis data lokal sumber Anda di DMS Fleet Advisor. Rekomendasi mencakup satu atau lebih kemungkinan mesin AWS target yang dapat Anda pilih untuk migrasi basis data lokal sumber Anda. Dari kemungkinan mesin target ini, DMS Fleet Advisor menyarankan mesin target tunggal sebagai tujuan migrasi berukuran tepat dan menunjukkan target ini seperti yang direkomendasikan DMS. Untuk menentukan tujuan migrasi berukuran tepat ini, DMS Fleet Advisor menggunakan metadata inventaris dan metrik yang dikumpulkan oleh pengumpul data Anda.

Anda dapat menggunakan rekomendasi sebelum memulai migrasi untuk menemukan opsi migrasi, menghemat biaya, dan mengurangi risiko. Anda dapat mengekspor rekomendasi sebagai file nilai dipisahkan koma (CSV), dan membagikannya dengan pemangku kepentingan utama untuk memfasilitasi pengambilan keputusan. Anda dapat mengekspor rekomendasi ke dalam AWS Pricing Calculator untuk lebih mengoptimalkan biaya pemeliharaan. Untuk informasi lebih lanjut, lihat <https://calculator.aws/#/>.

Anda tidak dapat mengubah rekomendasi target di DMS Fleet Advisor. Dengan demikian, Anda tidak dapat menggunakan DMS Fleet Advisor untuk analisis bagaimana-jika. Analisis bagaimana-jika adalah proses mengubah parameter target untuk melihat bagaimana perubahan tersebut memengaruhi estimasi harga rekomendasi Anda. Anda dapat menjalankan analisis bagaimana-jika AWS Pricing Calculator menggunakan parameter target yang direkomendasikan sebagai titik awal dalam. AWS Pricing Calculator Untuk informasi lebih lanjut, lihat <https://calculator.aws/#/>.

Kami menyarankan Anda mempertimbangkan rekomendasi DMS Fleet Advisor sebagai titik awal dalam perencanaan migrasi Anda. Anda kemudian dapat memutuskan untuk mengubah parameter instans yang direkomendasikan untuk mengoptimalkan biaya atau kinerja beban kerja database Anda.

Topik

- [Contoh target yang direkomendasikan](#)
- [Bagaimana cara DMS Fleet Advisor menentukan spesifikasi instans target untuk rekomendasi?](#)
- [Menghasilkan rekomendasi target dengan AWS DMS Fleet Advisor](#)
- [Menjelajahi detail rekomendasi target dengan AWS DMS Fleet Advisor](#)
- [Mengekspor rekomendasi target dengan AWS DMS Fleet Advisor](#)

- [Menemukan dan menganalisis batasan migrasi dengan AWS DMS Fleet Advisor](#)
- [Pemecahan masalah untuk rekomendasi target](#)

Contoh target yang direkomendasikan

Untuk rekomendasi target, DMS Fleet Advisor mempertimbangkan instans Amazon RDS DB untuk tujuan umum, yang dioptimalkan untuk memori, dan kinerja burstable berikut.

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

Untuk informasi selengkapnya tentang kelas instans Amazon RDS DB, lihat [kelas instans DB](#) di Panduan Pengguna Amazon RDS.

Bagaimana cara DMS Fleet Advisor menentukan spesifikasi instans target untuk rekomendasi?

DMS Fleet Advisor dapat menghasilkan rekomendasi berdasarkan kapasitas database atau pemanfaatan.

- Jika Anda memilih untuk membuat rekomendasi berdasarkan kapasitas database, maka DMS Fleet Advisor memetakan kapasitas database yang ada ke spesifikasi kelas instance terdekat.
- Jika Anda memilih untuk membuat rekomendasi berdasarkan pemanfaatan sumber daya, maka DMS Fleet Advisor menentukan nilai persentil ke-95 untuk metrik seperti CPU, memori, throughput IO, dan IOPS. Persentil ke-95 berarti bahwa 95 persen data yang dikumpulkan lebih rendah dari nilai ini. Kemudian, DMS Fleet Advisor memetakan nilai-nilai ini ke spesifikasi kelas instance terdekat.

Untuk menentukan ukuran basis data target, DMS Fleet Advisor mengumpulkan informasi tentang ukuran basis data sumber Anda. Kemudian, DMS Fleet Advisor merekomendasikan penggunaan ukuran yang sama untuk penyimpanan target. Jika penyimpanan basis data sumber Anda dilebih-lebihkan, maka ukuran penyimpanan target yang disarankan juga akan dilebih-lebihkan.

Jika Anda ingin memigrasikan data menggunakan AWS DMS, Anda mungkin perlu meningkatkan penyediaan IOPS untuk instans DB target Anda. Ketika DMS Fleet Advisor menghasilkan rekomendasi target, layanan hanya mempertimbangkan metrik basis data sumber Anda. DMS Fleet Advisor tidak mempertimbangkan IOPS tambahan yang mungkin Anda perlukan untuk menjalankan tugas migrasi data. Untuk informasi selengkapnya, lihat [Tugas migrasi berjalan perlahan](#).

Untuk memperkirakan biaya IOPS, DMS Fleet Advisor menggunakan one-to-one pemetaan penggunaan IOPS sumber Anda sebagai dasar. DMS Fleet Advisor menganggap beban puncak sebagai nilai dasar dan pemanfaatan 100% untuk harga IOPS.

Untuk database sumber PostgreSQL dan MySQL, DMS Fleet Advisor dapat menyertakan instans Aurora dan Amazon RDS DB dalam rekomendasi target. Jika konfigurasi Aurora memetakan ke persyaratan sumber, maka DMS Fleet Advisor menandai opsi ini seperti yang direkomendasikan.

Menghasilkan rekomendasi target dengan AWS DMS Fleet Advisor

Setelah Anda menyelesaikan pengumpulan data dan inventaris database dan armada analitik Anda, Anda dapat menghasilkan rekomendasi target di DMS Fleet Advisor. Untuk melakukannya, pilih database sumber dan konfigurasi pengaturan yang digunakan fitur Rekomendasi Target Penasihat Armada DMS untuk menentukan ukuran instance target. Selain itu, fitur Rekomendasi Target Penasihat Armada DMS menggunakan metrik kapasitas dan pemanfaatan yang dikumpulkan dari basis data sumber Anda.

Untuk menghasilkan rekomendasi target

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.

Pastikan Anda memilih Wilayah AWS tempat Anda menggunakan DMS Fleet Advisor.

2. Di panel navigasi, pilih Rekomendasi di bawah Menilai, lalu pilih Hasilkan rekomendasi.
3. Di panel Pilih database sumber, pilih kotak centang untuk nama database yang ingin Anda migrasi ke. AWS Cloud

Untuk database sumber pencarian, masukkan nama database Anda untuk memfilter inventaris Anda.

DMS Fleet Advisor dapat menghasilkan rekomendasi hingga 100 database sekaligus.

4. Untuk Ketersediaan dan daya tahan, pilih opsi penerapan yang disukai.

Untuk menghitung rekomendasi target untuk basis data produksi Anda, pilih Produksi (Multi-AZ). DMS Fleet Advisor menyertakan dua instans DB di Availability Zone yang berbeda dalam rekomendasi target Anda. Opsi penyebaran Multi-AZ ini memberikan ketersediaan tinggi, redundansi data, dan dukungan failover.

Jika Aurora adalah mesin target yang direkomendasikan dan Availability and Durability adalah penerapan Multi-AZ, rekomendasi target mencakup instans DB pembaca dan penulis.

Untuk menghitung rekomendasi target untuk database yang Anda gunakan untuk pengembangan atau pengujian, pilih Dev/Test (Single-AZ). DMS Fleet Advisor menyertakan satu instans DB dalam rekomendasi target Anda. Opsi penyebaran Single-AZ ini mengurangi biaya pemeliharaan.

5. Untuk ukuran instans Target, pilih opsi pilihan yang digunakan DMS Fleet Advisor untuk menghitung rekomendasi target.

Untuk menghitung rekomendasi target berdasarkan database sumber atau konfigurasi server OS Anda, pilih Total kapasitas. DMS Fleet Advisor menggunakan metrik seperti total CPU, memori, dan kapasitas disk database sumber atau server OS Anda untuk menghasilkan rekomendasi target. Kemudian, DMS Fleet Advisor memetakan metrik kapasitas database Anda ke spesifikasi kelas instans Amazon RDS DB terdekat.

Untuk menghitung rekomendasi target berdasarkan pemanfaatan sebenarnya dari basis data sumber atau server OS Anda, pilih Pemanfaatan sumber daya. DMS Fleet Advisor menggunakan metrik pemanfaatan CPU, memori, dan kapasitas disk database sumber atau server OS Anda untuk menghasilkan rekomendasi target. Dari metrik pemanfaatan, DMS Fleet Advisor menghitung persentil ke-95 untuk setiap metrik. Persentil ke-95 berarti bahwa 95 persen data dalam periode tersebut lebih rendah dari nilai ini. Kemudian, DMS Fleet Advisor memetakan nilai-nilai ini ke kelas instans Amazon RDS DB terdekat.

Kami menyarankan Anda menggunakan opsi pemanfaatan Sumber Daya untuk rekomendasi yang lebih akurat. Untuk melakukannya, pastikan Anda telah mengumpulkan total kapasitas dan metrik pemanfaatan sumber daya.

6. Pilih Hasilkan.

DMS Fleet Advisor menghasilkan rekomendasi target untuk database yang dipilih. Untuk rekomendasi yang berhasil dihasilkan, DMS Fleet Advisor menetapkan status ke Computed. Juga, DMS Fleet Advisor menggunakan AWS Pricing Calculator untuk menentukan perkiraan biaya bulanan untuk instans DB target yang direkomendasikan. Sekarang, Anda dapat menjelajahi rekomendasi yang dihasilkan secara rinci. Untuk informasi selengkapnya, lihat [Rincian rekomendasi](#).

Untuk memperkirakan total biaya bulanan untuk inventaris data Anda, pilih kotak centang untuk database yang Anda rencanakan untuk dipindahkan ke cloud. DMS Fleet Advisor menampilkan total perkiraan biaya bulanan dan ringkasan database target Anda di AWS Cloud DMS Fleet Advisor menggunakan Daftar Harga AWS Query API untuk memberikan detail harga untuk informasi Anda saja. Biaya aktual Anda tergantung pada berbagai faktor, termasuk penggunaan aktual Anda Layanan AWS. Untuk informasi selengkapnya tentang Layanan AWS harga, lihat [Harga Layanan Cloud](#).

Menjelajahi detail rekomendasi target dengan AWS DMS Fleet Advisor

Setelah DMS Fleet Advisor menghasilkan rekomendasi target, Anda dapat melihat parameter utama dari target migrasi yang direkomendasikan di tabel Rekomendasi. Parameter kunci ini termasuk mesin target, kelas instance, jumlah CPU virtual, memori, penyimpanan, dan jenis penyimpanan. Selain parameter ini, DMS Fleet Advisor menampilkan perkiraan biaya bulanan dari target migrasi yang direkomendasikan ini.

Setiap rekomendasi mungkin mencakup satu atau lebih mesin AWS target yang mungkin. Jika rekomendasi Anda mencakup beberapa mesin target, maka AWS DMS tandai salah satunya seperti yang direkomendasikan. Juga, AWS DMS menampilkan parameter dan perkiraan biaya bulanan untuk opsi yang direkomendasikan ini di tabel Rekomendasi.

Untuk membandingkan rekomendasi target dengan pemanfaatan dan kapasitas basis data sumber Anda, jelajahi rekomendasi Anda secara rinci. Selain itu, Anda dapat melihat batasan migrasi untuk rekomendasi yang dipilih. Batasan ini mencakup fitur database yang tidak didukung, item tindakan, dan pertimbangan migrasi lainnya.

Untuk mengeksplorasi rekomendasi secara rinci

1. Hasilkan rekomendasi target dengan DMS Fleet Advisor. Untuk informasi selengkapnya, lihat [Menghasilkan rekomendasi target](#).
2. Pilih nama rekomendasi dari tabel Rekomendasi. Halaman rekomendasi terbuka.

3. Jika rekomendasi Anda mencakup lebih dari satu opsi target, maka untuk rekomendasi Target, pilih opsi target.
4. Perluas bagian pemanfaatan dan kapasitas Sumber. DMS Fleet Advisor menampilkan bagan pemanfaatan sumber daya untuk metrik berikut.
 - Jumlah CPU
 - Memori
 - I/O throughput
 - Operasi input/output per detik (IOPS)
 - Penyimpanan
 - Jumlah koneksi server database aktif

Gunakan bagan ini untuk membandingkan metrik basis data sumber Anda dari pengumpul data DMS Anda dengan metrik mesin target yang dipilih.

Jika Anda tidak dapat melihat bagan setelah memperluas bagian Pemanfaatan dan kapasitas Sumber, pastikan Anda memberikan izin kepada pengguna IAM untuk melihat dasbor Amazon CloudWatch Untuk informasi selengkapnya, lihat [Menggunakan CloudWatch dasbor Amazon](#) di Panduan CloudWatch Pengguna Amazon.

5. Pilih tautan dengan nama mesin target yang Anda pilih. Halaman detail Target terbuka.
6. Untuk mengekspor rekomendasi target ke CSV, pilih opsi Ekspor ke CSV dari dropdown Tindakan.
7. Untuk mengekspor rekomendasi target AWS Pricing Calculator, pilih Optimalkan biaya dengan AWS Pricing Calculator opsi dari dropdown Tindakan.
8. Di bagian Konfigurasi, bandingkan nilai parameter basis data sumber Anda dengan parameter mesin target. Untuk mesin target, DMS Fleet Advisor menampilkan perkiraan biaya bulanan untuk sumber daya cloud Anda. DMS Fleet Advisor menggunakan Daftar Harga AWS Query API untuk memberikan detail harga untuk informasi Anda saja. Biaya aktual Anda tergantung pada berbagai faktor, termasuk penggunaan aktual Anda Layanan AWS. Untuk informasi selengkapnya tentang Layanan AWS harga, lihat Harga Layanan <https://aws.amazon.com/pricing/> Cloud.
9. Di bagian Batasan migrasi, lihat batasan migrasi. Kami menyarankan Anda mempertimbangkan batasan ini saat memigrasikan basis data sumber Anda ke AWS Cloud

Mengekspor rekomendasi target dengan AWS DMS Fleet Advisor

Setelah Anda membuat rekomendasi target, Anda dapat menyimpan salinan daftar rekomendasi sebagai file nilai dipisahkan koma (CSV).

Untuk menghasilkan rekomendasi target

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.

Pastikan Anda memilih Wilayah AWS tempat Anda menggunakan DMS Fleet Advisor.

2. Di panel navigasi, pilih Rekomendasi di bawah Nilai, lalu pilih rekomendasi yang akan disertakan dalam file CSV Anda.
3. Pilih Ekspor ke CSV, masukkan nama file dan pilih folder di PC Anda tempat menyimpan file ini.
4. Buka file CSV.

File CSV dengan rekomendasi berisi informasi berikut.

- **CreatedDate**— Tanggal ketika DMS Fleet Advisor membuat rekomendasi mesin target.
- **Databaseld**— Pengidentifikasi database sumber tempat DMS Fleet Advisor membuat rekomendasi ini.
- **DeploymentOption**— Opsi penerapan untuk instans Amazon RDS DB yang direkomendasikan.
- **EngineEdition**— Target yang direkomendasikan edisi mesin Amazon RDS.
- **EngineName**— Nama mesin target.
- **InstanceMemory**— Jumlah memori pada instans Amazon RDS DB yang direkomendasikan.
- **InstanceSizingType**— Ukuran instance target Anda.
- **InstanceType**— Jenis instans Amazon RDS target yang direkomendasikan.
- **InstanceVcpu**— Jumlah CPU virtual pada instans Amazon RDS DB yang direkomendasikan.
- **Preferred** - Bendera Boolean yang menunjukkan bahwa opsi target ini direkomendasikan.
- **Status** — Status rekomendasi mesin target.
- **Storagelops**— Jumlah operasi I/O yang diselesaikan setiap detik (IOPS) pada instans Amazon RDS DB yang direkomendasikan.
- **StorageSize**— Ukuran penyimpanan instans Amazon RDS DB yang direkomendasikan.
- **StorageType**— Jenis penyimpanan instans Amazon RDS DB yang direkomendasikan.

- **WorkloadType**— Opsi penyebaran untuk mesin target Anda seperti penyebaran multi-AZ atau Single-AZ.

Menemukan dan menganalisis batasan migrasi dengan AWS DMS Fleet Advisor

Anda dapat menggunakan pengumpul data DMS untuk menemukan fitur database yang tidak didukung oleh mesin target Anda. Untuk memilih target migrasi yang tepat, Anda harus mempertimbangkan batasan ini.

Pengumpul data DMS menemukan fitur database sumber tertentu. Kemudian, DMS Fleet Advisor menganalisis fitur sumber dari sudut pandang migrasi ke target yang ditentukan dan memberikan informasi tambahan tentang batasan tersebut dan mencakup tindakan yang disarankan untuk mengatasi atau menghindari batasan ini. Juga, DMS Fleet Advisor menghitung dampak dari keterbatasan ini.

Daftar batasan tersedia di halaman detail mesin Target. Arahkan ke halaman ini dari halaman Rekomendasi di menu navigasi kiri. Dari daftar target, pilih mesin target untuk diperiksa. Daftar batasan ada di bagian bawah halaman.

Tabel berikut mencakup fitur database MySQL yang tidak didukung Amazon RDS for MySQL.

Batasan	Deskripsi	Dampak
Plugin otentikasi	Amazon RDS tidak mendukung plugin otentikasi MySQL.	Rendah
Kesalahan saat masuk ke log sistem	Amazon RDS tidak mendukung penulisan log kesalahan ke log sistem.	Rendah
Pengidentifikasi transaksi global	Anda dapat menggunakan pengidentifikasi transaksi global dengan semua RDS untuk versi MySQL 5.7, dan RDS untuk MySQL versi	Rendah

Batasan	Deskripsi	Dampak
	8.0.26 dan versi MySQL 8.0 yang lebih tinggi.	
Replikasi Grup	Amazon RDS tidak mendukung plugin Replikasi Grup MySQL.	Rendah
Enkripsi tablespace InnoDB	Amazon RDS tidak mendukung enkripsi tablespace InnoDB.	Rendah
Kata yang dicadangkan InnoDB	InnoDB adalah kata yang dicadangkan untuk Amazon RDS for MySQL. Anda tidak dapat menggunakan nama ini untuk basis data MySQL.	Rendah
Plugin keyring	Amazon RDS tidak mendukung plugin keyring MySQL.	Rendah
Plugin validasi kata sandi	Amazon RDS tidak mendukung plugin <code>MySQLvalidate_password</code> .	Rendah
Variabel sistem yang dipertahankan	Amazon RDS tidak mendukung variabel sistem MySQL yang bertahan.	Rendah

Batasan	Deskripsi	Dampak
Akses terbatas	Amazon RDS membatasi akses ke prosedur dan tabel sistem tertentu yang memerlukan izin lanjutan. Selain itu, Amazon RDS tidak mengizinkan akses host langsung ke instans DB dengan menggunakan Telnet, Secure Shell (SSH), atau Windows Remote Desktop Connection.	Rendah
Plugin tulis ulang kueri penulis ulang	Amazon RDS tidak mendukung plugin penulisan ulang kueri MySQL Rewriter.	Rendah
Replikasi semisinkron	Amazon RDS tidak mendukung replikasi semisinkron MySQL.	Rendah
Ruang Meja yang Dapat Diangkut	Amazon RDS tidak mendukung fitur MySQL Transportable Tablespaces.	Rendah
Plugin X	Amazon RDS tidak mendukung Plugin MySQL X.	Rendah

Tabel berikut mencakup fitur database Oracle yang Amazon RDS for Oracle tidak mendukung.

Batasan	Deskripsi	Dampak
Penjaga Data Aktif	Anda tidak dapat menggunakan Active Data Guard dengan database kontainer multitenant Oracle (CDB).	Sedang

Batasan	Deskripsi	Dampak
Manajemen Penyimpanan Otomatis	Amazon RDS tidak mendukung Oracle Automatic Storage Management (Oracle ASM).	Sedang
Stream Aktivitas Basis Data	Amazon RDS tidak mendukung Oracle Database Activity Streams untuk arsitektur penyewa tunggal.	Tinggi
Batas ukuran file	Ukuran maksimum satu file pada RDS untuk instans Oracle DB adalah 16 TiB.	Sedang
FTP dan SFTP	Amazon RDS tidak mendukung FTP dan SFTP.	Sedang
Tabel partisi hibrida	Amazon RDS tidak mendukung tabel partisi hibrida Oracle.	Tinggi
Oracle Data Guard	Amazon RDS bukan Oracle Data Guard untuk arsitektur penyewa tunggal.	Tinggi
Brankas Database Oracle	Amazon RDS tidak mendukung Oracle Database Vault.	Tinggi
Hak istimewa Oracle DBA Vault	Amazon RDS memiliki batasan untuk hak istimewa Oracle DBA. Untuk informasi selengkapnya, lihat Batasan untuk hak istimewa Oracle DBA .	Tinggi

Batasan	Deskripsi	Dampak
Oracle Enterprise Manager	Amazon RDS bukan Oracle Enterprise Manager untuk arsitektur penyewa tunggal.	Tinggi
Agen Manajer Perusahaan Oracle	Amazon RDS bukan Agen Manajer Oracle Enterprise untuk arsitektur penyewa tunggal.	Sedang
Repositori Manajemen Oracle Enterprise Manager Cloud Control	Anda tidak dapat menggunakan instan Amazon RDS for Oracle DB untuk Oracle Enterprise Manager Cloud Control Management Repository.	Tinggi
Database Kilas Balik Oracle	Amazon RDS tidak mendukung fitur Oracle Flashback Database.	Tinggi
Keamanan Label Oracle	Amazon RDS tidak mendukung Oracle Label Security untuk arsitektur penyewa tunggal. Anda dapat menggunakan Oracle Label Security hanya dengan database kontainer multitenant (Oracle CDB).	Tinggi
Gerbang Pesan Oracle	Amazon RDS tidak mendukung Oracle Messaging Gateway.	Tinggi

Batasan	Deskripsi	Dampak
Cluster Aplikasi Nyata Oracle	Amazon RDS tidak mendukung Oracle Real Application Clusters (Oracle RAC).	Tinggi
Pengujian Aplikasi Nyata Oracle	Amazon RDS tidak mendukung Oracle Real Application Testing.	Tinggi
Database Oracle Snapshot Standby	Amazon RDS tidak mendukung database Oracle Snapshot Standby.	Tinggi
Sinonim publik	Amazon RDS tidak mendukung sinonim publik untuk skema yang disediakan Oracle.	Sedang
Skema untuk fitur yang tidak didukung	Amazon RDS tidak mendukung skema untuk fitur Oracle dan komponen yang memerlukan hak istimewa sistem.	Tinggi
Audit terpadu murni	Amazon RDS tidak mendukung audit terpadu murni. Anda dapat menggunakan audit terpadu dalam mode campuran.	Sedang
Manajer Ruang Kerja	Amazon RDS tidak mendukung skema Oracle Database Workspace Manager. WMSYS	Tinggi

Tabel berikut mencakup fitur database PostgreSQL yang tidak didukung Amazon RDS for PostgreSQL.

Batasan	Deskripsi	Dampak
Koneksi bersamaan	Jumlah maksimum koneksi bersamaan ke RDS Anda untuk instance PostgreSQL dibatasi oleh parameter. <code>max_connections</code>	Rendah
Versi terbaru	Amazon RDS tidak menerapkan peningkatan versi utama secara otomatis. Untuk melakukan upgrade versi utama, modifikasi instans DB Anda secara manual. Untuk informasi selengkapnya, lihat Memilih upgrade versi utama untuk PostgreSQL .	Rendah
Koneksi yang dipesan	Amazon RDS mencadangkan hingga 3 koneksi untuk pemeliharaan sistem. Jika Anda menentukan nilai untuk parameter koneksi pengguna, tambahkan 3 ke jumlah koneksi yang Anda harapkan untuk digunakan.	Rendah
Ekstensi yang didukung	RDS untuk PostgreSQL mendukung sejumlah ekstensi terbatas untuk mesin database PostgreSQL. Anda dapat menemukan daftar ekstensi yang didukung dalam grup parameter DB default untuk versi PostgreSQL Anda. Anda	Rendah

Batasan	Deskripsi	Dampak
	juga dapat melihat daftar ekstensi saat ini menggunakan <code>psql</code> dengan menunjukkan <code>rds.extensions</code> parameter.	
Pemisahan atau isolasi tablespace	Anda tidak dapat menggunakan ruang tabel untuk pemisahan atau isolasi I/O. Dalam RDS untuk PostgreSQL, semua penyimpanan berada pada volume logis tunggal.	Rendah

Tabel berikut mencakup fitur database SQL Server yang tidak didukung Amazon RDS for SQL Server.

Batasan	Deskripsi	Dampak
Mencadangkan ke Microsoft Azure Blob Storage	RDS untuk SQL Server tidak mendukung pencadangan ke Microsoft Azure Blob Storage.	Sedang
Ekstensi kumpulan buffer	RDS untuk SQL Server tidak mendukung ekstensi kumpulan buffer.	Tinggi
Kebijakan kata sandi khusus	RDS untuk SQL Server tidak mendukung kebijakan kata sandi khusus.	Sedang
Layanan Kualitas Data	RDS untuk SQL Server tidak mendukung SQL Server Data Quality Services (DQS).	Tinggi

Batasan	Deskripsi	Dampak
Pengiriman Log Basis Data	RDS untuk SQL Server tidak mendukung pengiriman log database.	Tinggi
Nama basis data	Nama database memiliki batasan berikut: tidak dapat memulai dengan rdsadmin; tidak dapat memulai atau mengakhiri dengan spasi atau tab; tidak dapat berisi karakter apa pun yang membuat baris baru; tidak dapat berisi tanda kutip tunggal (').	Sedang
Basis data snapshot	RDS untuk SQL Server tidak mendukung snapshot database. Anda hanya dapat menggunakan snapshot instans DB di Amazon RDS.	Sedang
Prosedur tersimpan yang diperluas	RDS untuk SQL Server tidak mendukung prosedur tersimpan yang diperluas, termasuk. xp_cmdshe11	Tinggi
Tabel file	RDS untuk SQL Server tidak mendukung tabel file.	Sedang
Dukungan FILESTREAM	RDS untuk SQL Server tidak menyediakan dukungan FILESTREAM.	Sedang
Server tertaut	RDS untuk SQL Server menyediakan dukungan terbatas untuk server tertaut.	Tinggi

Batasan	Deskripsi	Dampak
Machine Learning dan Layanan R	RDS untuk SQL Server tidak mendukung Machine Learning dan R Services karena Anda memerlukan akses OS untuk menginstal layanan ini.	Tinggi
Rencana pemeliharaan	RDS untuk SQL Server tidak mendukung rencana pemeliharaan.	Tinggi
Pengumpul Data Performa	RDS untuk SQL Server tidak mendukung Performance Data Collector.	Tinggi
Manajemen Berbasis Kebijakan	RDS untuk SQL Server tidak mendukung Manajemen Berbasis Kebijakan.	Sedang
PolyBase	RDS untuk SQL Server tidak mendukung PolyBase	Tinggi
Replikasi	RDS untuk SQL Server tidak mendukung replikasi.	Sedang
Resource Governor	RDS untuk SQL Server tidak mendukung Resource Governor.	Tinggi
Pemicu tingkat server	RDS untuk SQL Server tidak mendukung pemicu tingkat server.	Sedang
Titik Akhir Service Broker	RDS untuk SQL Server tidak mendukung titik akhir Service Broker.	Tinggi

Batasan	Deskripsi	Dampak
SSAS	Pertimbangkan batasan yang berlaku untuk menjalankan SQL Server Analysis Services (SSAS) pada RDS untuk SQL Server. Untuk informasi selengkapnya, lihat Batas .	Rendah
SSIS	Pertimbangkan batasan yang berlaku untuk menjalankan SQL Server Integration Services (SSIS) pada RDS untuk SQL Server. Untuk informasi selengkapnya, lihat Batas .	Rendah
SSRS	Pertimbangkan batasan yang berlaku untuk menjalankan SQL Server Reporting Services (SSRS) pada RDS untuk SQL Server. Untuk informasi selengkapnya, lihat Batas .	Rendah
Ukuran penyimpanan untuk instans SQL Server DB	<p>Ukuran penyimpanan maksimum untuk penyimpanan SQL Server General Purpose (SSD) dan instans penyimpanan IOPS Provisioned adalah 16 TiB.</p> <p>Ukuran penyimpanan maksimum untuk instans penyimpanan SQL Server Magnetic adalah 1 TiB.</p>	Tinggi

Batasan	Deskripsi	Dampak
Database Peregangan	RDS untuk SQL Server tidak mendukung fitur SQL Server Stretch Database.	Sedang
Titik akhir T-SQL	RDS untuk SQL Server tidak mendukung semua operasi yang digunakan. CREATE ENDPOINT	Tinggi
Properti database YANG DAPAT DIPERCAYA	RDS untuk SQL Server tidak mendukung properti TRUSTWORTHY database karena memerlukan peran. sysadmin	Sedang

Tabel berikut mencakup daftar masalah rekomendasi. DMS Fleet Advisor menganalisis sumber dan fitur basis data target dan menyediakan batasan migrasi ini. Batasan dengan dampak Blocker berarti bahwa DMS Fleet Advisor tidak dapat menghasilkan rekomendasi target untuk database sumber.

Batasan	Deskripsi	Dampak
Contoh yang tepat tidak ditemukan	AWS DMStidak dapat menemukan instance target yang dapat berfungsi sebagai tujuan migrasi berukuran tepat untuk kombinasi metrik basis data sumber Anda.	Pemblokir
Contoh yang tepat tidak ditemukan oleh IOPS	Basis data sumber menggunakan sejumlah IOPS, yang melebihi jumlah maksimum IOPS untuk kemungkinan instans DB target.	Pemblokir

Batasan	Deskripsi	Dampak
Contoh yang tepat tidak ditemukan oleh RAM	Database sumber menggunakan sejumlah GB RAM, yang melebihi ukuran maksimum RAM untuk kemungkinan instans DB target.	Pemblokir
Contoh yang tepat tidak ditemukan berdasarkan ukuran penyimpanan	Database sumber menggunakan sejumlah TB penyimpanan, yang melebihi ukuran penyimpanan maksimum untuk kemungkinan instans DB target.	Pemblokir
Contoh yang tepat tidak ditemukan oleh edisi	Database sumber memiliki edisi, yang tidak didukung oleh Amazon RDS.	Pemblokir
Contoh yang tepat tidak ditemukan oleh core CPU	Database sumber memiliki sejumlah inti CPU, yang melebihi jumlah maksimum inti CPU untuk kemungkinan instans DB target.	Pemblokir
Contoh yang tepat tidak ditemukan berdasarkan versi	Database sumber Anda memiliki versi, yang AWS DMS tidak dikenali.	Pemblokir

Batasan	Deskripsi	Dampak
Parameter CPU tidak terdefinisi	Pengumpul data DMS tidak mengumpulkan informasi tentang CPU yang digunakan basis data sumber Anda. Pastikan Anda mengumpulkan metrik yang diperlukan dan kredensi yang dikonfigurasi untuk penerusan data dalam pengumpul data Anda. Lihat Mengkonfigurasi kredensi untuk penerusan data .	Pemblokir
Parameter memori tidak terdefinisi	Pengumpul data DMS tidak mengumpulkan informasi tentang memori yang digunakan basis data sumber Anda. Pastikan Anda mengumpulkan metrik yang diperlukan dan kredensi yang dikonfigurasi untuk penerusan data dalam pengumpul data Anda. Lihat Mengkonfigurasi kredensi untuk penerusan data .	Pemblokir

Batasan	Deskripsi	Dampak
Parameter ukuran penyimpanan tidak terdefinisi	Pengumpul data DMS tidak mengumpulkan informasi tentang ukuran penyimpanan yang digunakan database sumber Anda. Pastikan Anda mengumpulkan metrik yang diperlukan dan kredensi yang dikonfigurasi untuk penerusan data dalam pengumpul data Anda. Lihat Mengkonfigurasi kredensi untuk penerusan data .	Pemblokir
Parameter IOPS penyimpanan tidak terdefinisi	Pengumpul data DMS tidak mengumpulkan metrik IOPS penyimpanan untuk penggunaan basis data sumber Anda. Pastikan Anda mengumpulkan metrik yang diperlukan dan kredensi yang dikonfigurasi untuk penerusan data dalam pengumpul data Anda.	Pemblokir
Tidak Cukup Data	Pengumpul data DMS tidak mengumpulkan cukup data untuk menghasilkan rekomendasi target. Pastikan Anda mengonfigurasi kredensial untuk penerusan data di pengumpul data Anda. Lihat Mengkonfigurasi kredensi untuk penerusan data .	Pemblokir

Batasan	Deskripsi	Dampak
Edisi database tidak terdefinisi	Pengumpul data DMS tidak mengumpulkan informasi tentang edisi basis data sumber Anda. Pastikan Anda mengumpulkan metrik yang diperlukan dan kredensi yang dikonfigurasi untuk penerusan data dalam pengumpul data Anda. Lihat Mengkonfigurasi kredensi untuk penerusan data .	Pemblokir
Kesalahan Tidak Diketahui	DMS Fleet Advisor tidak dapat menghasilkan rekomendasi target untuk basis data sumber Anda.	Pemblokir
Versi database tidak terdefinisi	Penasihat Armada DMS tidak mengumpulkan informasi tentang versi basis data sumber Anda. DMS Fleet Advisor merekomendasikan agar Anda menggunakan versi database terbaru untuk basis data sumber Anda. Jika Anda memilih rekomendasi ini, maka Anda harus meng-upgrade versi database Anda. Tinjau rekomendasi target yang dihasilkan untuk basis data sumber Anda dan pastikan bahwa rekomendasi ini memenuhi persyaratan Anda.	Tinggi

Batasan	Deskripsi	Dampak
Meningkatkan jumlah koneksi database dalam pengaturan RDS	Database sumber Anda memerlukan sejumlah koneksi tertentu. Secara default, jumlah koneksi yang tersedia untuk instans database Amazon RDS berbeda. Pastikan bahwa Anda mengubah nilai default ini ketika Anda membuat instance database RDS Anda. Untuk melakukannya, perbarui nilai <code>max_connections</code> parameter di Grup Parameter.	Sedang
Edisi target kompatibel	Rekomendasi target untuk database sumber Anda menggunakan edisi database yang berbeda. Edisi basis data sumber Anda mendukung fitur yang sama dengan edisi target yang direkomendasikan. Namun, memilih edisi database baru ini dapat meningkatkan pengeluaran Anda.	Sedang

Batasan	Deskripsi	Dampak
Parameter throughput penyimpanan tidak terdefinisi	Pengumpul data DMS tidak mengumpulkan metrik throughput penyimpanan untuk penggunaan basis data sumber Anda. Tinjau rekomendasi target yang dihasilkan untuk basis data sumber Anda dan pastikan bahwa rekomendasi ini memenuhi persyaratan Anda.	Sedang
Parameter nomor koneksi database tidak terdefinisi	Pengumpul data DMS tidak mengumpulkan informasi tentang jumlah koneksi yang digunakan basis data sumber Anda. Tinjau rekomendasi target yang dihasilkan untuk basis data sumber Anda dan pastikan bahwa rekomendasi ini memenuhi persyaratan Anda. Atau, minta kenaikan kuota.	Sedang
Versi downgrade database	Database sumber Anda berjalan pada versi yang lebih tinggi daripada database Amazon RDS. Untuk menurunkan versi database Anda, pastikan Anda tidak menggunakan fitur yang tidak diterapkan dalam versi yang lebih rendah. Atau, gunakan Amazon EC2 sebagai target migrasi.	Sedang

Batasan	Deskripsi	Dampak
Edisi target berbeda	Rekomendasi target untuk database sumber Anda menggunakan edisi database yang berbeda. Edisi basis data sumber Anda kompatibel dengan edisi target yang direkomendasikan. Namun, edisi basis data target yang disarankan tidak mendukung beberapa fitur edisi basis data sumber Anda. Memilih edisi database baru ini dapat meningkatkan pengeluaran Anda.	Sedang

Batasan	Deskripsi	Dampak
Upgrade dari versi yang tidak didukung	<p>Database sumber Anda telah mencapai akhir tahap dukungan. Untuk menggunakan versi mesin DB terbaru sebagai target, tingkatkan database Anda sebelum migrasi. Atau, gunakan Amazon EC2 sebagai target migrasi.</p> <p>Bergantung pada mesin database, gunakan salah satu tautan berikut untuk Pelajari lebih lanjut:</p> <p>Memutakhirkan MySQL</p> <p>Tingkatkan SQL Server</p> <p>Tingkatkan OracleDB</p> <p>Tingkatkan PostgreSQL</p>	Sedang

Pemecahan masalah untuk rekomendasi target

Dalam daftar berikut, Anda dapat menemukan tindakan yang harus diambil ketika Anda mengalami masalah dengan fitur Rekomendasi Target Penasihat Armada DMS.

Topik

- [Saya tidak dapat melihat perkiraan harga untuk rekomendasi target](#)
- [Saya tidak dapat melihat bagan pemanfaatan sumber daya](#)
- [Saya tidak dapat melihat status pengumpulan metrik](#)

Saya tidak dapat melihat perkiraan harga untuk rekomendasi target

Jika Anda melihat Tidak ada data untuk Estimasi biaya bulanan untuk rekomendasi dengan status Sukses, maka pastikan bahwa Anda memberi pengguna IAM Anda izin untuk mengakses API Daftar Harga AWS Layanan. Untuk melakukannya, Anda harus membuat kebijakan yang menyertakan `pricing:GetProducts` izin dan menambahkannya ke pengguna IAM Anda seperti yang dijelaskan dalam [Buat sumber daya IAM](#).

DMS Fleet Advisor tidak menghitung perkiraan biaya bulanan untuk rekomendasi dengan status Gagal.

Saya tidak dapat melihat bagan pemanfaatan sumber daya

Jika Anda melihat pesan Gagal memuat metrik setelah Anda memperluas bagian Pemanfaatan dan kapasitas Sumber, pastikan Anda memberikan izin kepada pengguna IAM untuk melihat dasbor Amazon CloudWatch. Untuk melakukannya, Anda harus menambahkan kebijakan yang diperlukan ke pengguna IAM Anda seperti yang dijelaskan dalam [Buat sumber daya IAM](#).

Atau, Anda dapat membuat kebijakan khusus yang mencakup `cloudwatch:GetDashboard`, `cloudwatch:ListDashboards`, `cloudwatch:PutDashboard`, dan `cloudwatch:DeleteDashboards` izin. Untuk informasi selengkapnya, lihat [Menggunakan CloudWatch dasbor Amazon](#) di Panduan CloudWatch Pengguna Amazon.

Saya tidak dapat melihat status pengumpulan metrik

Jika Anda melihat koleksi Tidak ada data yang tersedia untuk Metrik saat memilih Menghasilkan rekomendasi, pastikan Anda mengumpulkan data. Untuk informasi selengkapnya, lihat [Mengumpulkan data untuk AWS DMS Fleet Advisor](#).

Jika Anda mengalami masalah ini setelah mengumpulkan data, pastikan Anda memberikan `cloudwatch:Get*` izin kepada pengguna IAM untuk mengakses Amazon CloudWatch. DMS Fleet Advisor menggunakan peran terkait layanan untuk mempublikasikan metrik kinerja database yang dikumpulkan atas nama Anda. CloudWatch Pastikan untuk membuat peran terkait layanan untuk digunakan dengan DMS Fleet Advisor. Lihat informasi yang lebih lengkap di [Buat sumber daya IAM](#).

Batasan Penasihat Armada DMS

Batasan saat menggunakan DMS Fleet Advisor meliputi:

- DMS Fleet Advisor menghasilkan one-to-one rekomendasi. Untuk setiap basis data sumber, DMS Fleet Advisor menentukan mesin target tunggal. DMS Fleet Advisor tidak menangani server multitenant dan tidak memberikan rekomendasi untuk menjalankan beberapa database pada satu instans DB target.
- DMS Fleet Advisor tidak memberikan rekomendasi tentang peningkatan versi database yang tersedia.
- DMS Fleet Advisor menghasilkan rekomendasi hingga 100 database sekaligus.
- Jika Anda menginstal pengumpul data DMS, yang merupakan aplikasi Windows, pastikan Anda juga menginstal .NET Framework 4.8 dan PowerShell 6.0 dan yang lebih tinggi. Untuk persyaratan perangkat keras, lihat [Memasang pengumpul data](#).
- Pengumpul data DMS memerlukan izin untuk menjalankan permintaan menggunakan protokol LDAP di server domain Anda.
- Pengumpul data DMS membutuhkan skrip SSH sudo yang berjalan di Linux.
- Pengumpul data DMS memerlukan izin untuk menjalankan remote PowerShell, Windows Management Instrumentation (WMI), WMI Query Language (WQL), dan skrip registri di Windows.
- Untuk MySQL dan PostgreSQL, DMS Fleet Advisor tidak dapat mengumpulkan metrik kinerja dari database Anda. Sebagai gantinya, DMS Fleet Advisor mengumpulkan metrik server OS. Oleh karena itu, Anda tidak dapat membuat rekomendasi berdasarkan metrik pemanfaatan untuk database MySQL dan PostgreSQL yang berjalan di Amazon RDS dan Aurora.

Mengonversi skema database menggunakan Konversi Skema DMS

Konversi Skema DMS di AWS Database Migration Service (AWS DMS) membuat migrasi database antara berbagai jenis database lebih dapat diprediksi. Gunakan Konversi Skema DMS untuk menilai kompleksitas migrasi Anda untuk penyedia data sumber Anda, dan untuk mengonversi skema database dan objek kode. Anda kemudian dapat menerapkan kode yang dikonversi ke database target Anda.

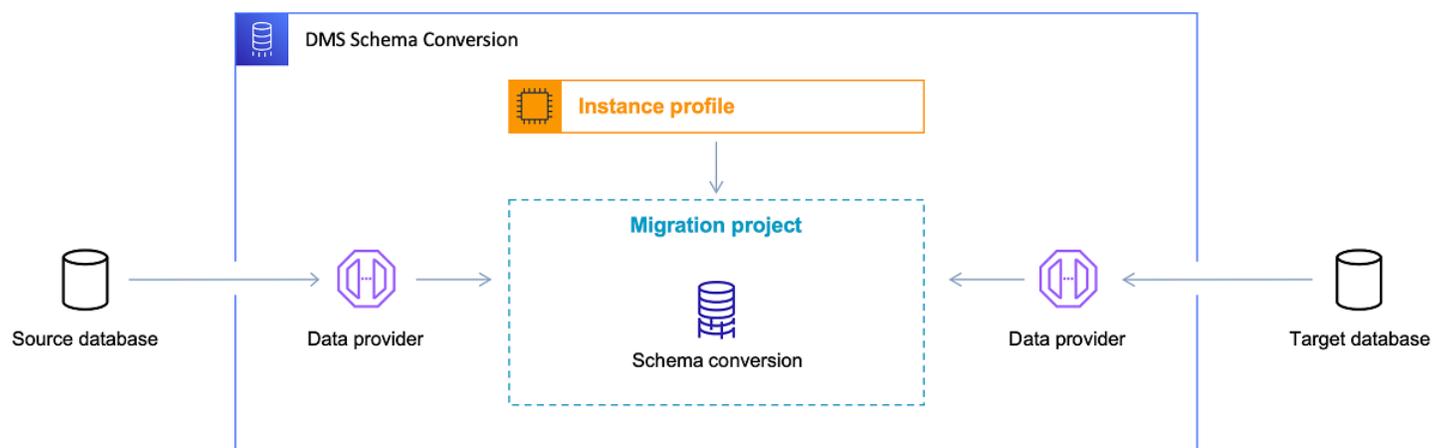
Konversi Skema DMS secara otomatis mengonversi skema basis data sumber Anda dan sebagian besar objek kode database ke format yang kompatibel dengan basis data target. Konversi ini mencakup tabel, tampilan, prosedur tersimpan, fungsi, tipe data, sinonim, dan sebagainya. Objek apa pun yang Konversi Skema DMS tidak dapat dikonversi secara otomatis ditandai dengan jelas. Untuk menyelesaikan migrasi, Anda dapat mengonversi objek ini secara manual.

Pada tingkat tinggi, [Konversi Skema DMS](#) beroperasi dengan tiga komponen berikut: profil instans, penyedia data, dan proyek migrasi. Profil instance menentukan pengaturan jaringan dan keamanan. Penyedia data menyimpan kredensial koneksi database. Proyek migrasi berisi penyedia data, profil instance, dan aturan migrasi. AWS DMS menggunakan penyedia data dan profil instance untuk merancang proses yang mengubah skema database dan objek kode.

Untuk daftar database sumber yang didukung, lihat [Sumber untuk Konversi Skema DMS](#).

Untuk daftar database target yang didukung, lihat [Target untuk Konversi Skema DMS](#).

Diagram berikut menggambarkan proses Konversi Skema DMS.



Gunakan topik berikut untuk lebih memahami cara menggunakan Konversi Skema DMS.

Topik

- [Didukung Wilayah AWS](#)
- [Fitur konversi skema](#)
- [Batasan konversi skema](#)
- [Memulai dengan Konversi Skema DMS](#)
- [Menyiapkan jaringan untuk Konversi Skema DMS](#)
- [Membuat penyedia data sumber dalam Konversi Skema DMS](#)
- [Membuat penyedia data target dalam Konversi Skema DMS](#)
- [Mengelola proyek migrasi dalam Konversi Skema DMS](#)
- [Membuat laporan penilaian migrasi database dengan Konversi Skema DMS](#)
- [Menggunakan Konversi Skema DMS](#)
- [Menggunakan paket ekstensi dalam Konversi Skema DMS](#)

Didukung Wilayah AWS

Anda dapat membuat proyek migrasi Konversi Skema DMS berikut ini. Wilayah AWS Di Wilayah lain, Anda dapat menggunakan AWS Schema Conversion Tool. Untuk informasi selengkapnya AWS SCT, lihat [Panduan Pengguna Alat Konversi AWS Skema](#).

Nama Wilayah	Wilayah
AS Timur (Virginia Utara)	us-east-1
US East (Ohio)	us-east-2
US West (Oregon)	us-west-2
Asia Pasifik (Tokyo)	ap-northeast-1
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Europe (Frankfurt)	eu-central-1

Nama Wilayah	Wilayah
Eropa (Stockholm)	eu-north-1
Eropa (Irlandia)	eu-west-1

Fitur konversi skema

Konversi Skema DMS menyediakan fitur-fitur berikut:

- Konversi Skema DMS secara otomatis mengelola AWS Cloud sumber daya yang diperlukan untuk proyek migrasi database Anda. Sumber daya ini termasuk profil instance, penyedia data, dan AWS Secrets Manager rahasia. Mereka juga menyertakan peran AWS Identity and Access Management (IAM), bucket Amazon S3, dan proyek migrasi.
- Anda dapat menggunakan Konversi Skema DMS untuk terhubung ke database sumber Anda, membaca metadata, dan membuat laporan penilaian migrasi database. Anda kemudian dapat menyimpan laporan ke ember Amazon S3. Dengan laporan ini, Anda mendapatkan ringkasan tugas konversi skema dan detail untuk item yang Konversi Skema DMS tidak dapat dikonversi secara otomatis ke basis data target Anda. Laporan penilaian migrasi database membantu mengevaluasi berapa banyak proyek migrasi Anda Konversi Skema DMS dapat diotomatisasi. Juga, laporan ini membantu memperkirakan jumlah upaya manual yang diperlukan untuk menyelesaikan konversi. Untuk informasi selengkapnya, lihat [Membuat laporan penilaian migrasi database dengan Konversi Skema DMS](#).
- Setelah Anda terhubung ke penyedia data sumber dan target Anda, Konversi Skema DMS dapat mengonversi skema basis data sumber yang ada ke mesin basis data target. Anda dapat memilih item skema apa pun dari database sumber Anda untuk dikonversi. Setelah Anda mengonversi kode database Anda dalam Konversi Skema DMS, Anda dapat meninjau kode sumber dan kode yang dikonversi. Selain itu, Anda dapat menyimpan kode SQL yang dikonversi ke bucket Amazon S3.
- Sebelum mengonversi skema basis data sumber, Anda dapat mengatur aturan transformasi. Anda dapat menggunakan aturan transformasi untuk mengubah tipe data kolom, memindahkan objek dari satu skema ke skema lainnya, dan mengubah nama objek. Anda dapat menerapkan aturan transformasi ke database, skema, tabel, dan kolom. Untuk informasi selengkapnya, lihat [Menyiapkan aturan transformasi](#).
- Anda dapat mengubah pengaturan konversi untuk meningkatkan kinerja kode yang dikonversi. Pengaturan ini spesifik untuk setiap pasangan konversi dan bergantung pada fitur database

sumber yang Anda gunakan dalam kode Anda. Untuk informasi selengkapnya, lihat [Menentukan pengaturan konversi skema](#).

- Dalam beberapa kasus, Konversi Skema DMS tidak dapat mengonversi fitur basis data sumber ke fitur Amazon RDS yang setara. Untuk kasus ini, Konversi Skema DMS membuat paket ekstensi di database target Anda untuk meniru fitur yang tidak dikonversi. Untuk informasi selengkapnya, lihat [Menggunakan paket ekstensi](#).
- Anda dapat menerapkan kode yang dikonversi dan skema paket ekstensi ke database target Anda. Untuk informasi selengkapnya, lihat [Menerapkan kode konversi Anda](#).
- Konversi Skema DMS mendukung semua fitur dalam rilis terbaru AWS SCT . Untuk informasi lebih lanjut, lihat [Catatan rilis terbaru untuk AWS SCT](#).
- Anda dapat mengedit kode SQL yang dikonversi sebelum DMS memigrasikannya ke database target. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan kode SQL Anda yang telah dikonversi](#) .

Batasan konversi skema

Konversi Skema DMS adalah versi web dari (). AWS Schema Conversion Tool AWS SCT Konversi Skema DMS mendukung lebih sedikit platform basis data dan menyediakan fungsionalitas yang lebih terbatas dibandingkan dengan aplikasi AWS SCT desktop. Untuk mengonversi skema gudang data, kerangka kerja data besar, kode SQL aplikasi, dan proses ETL, gunakan AWS SCT Untuk informasi selengkapnya AWS SCT, lihat [Panduan Pengguna Alat Konversi AWS Skema](#).

Batasan berikut berlaku saat Anda menggunakan Konversi Skema DMS untuk konversi skema database:

- Anda tidak dapat menyimpan proyek migrasi dan menggunakannya dalam mode offline.
- Anda tidak dapat mengedit kode SQL untuk sumber dalam proyek migrasi untuk Konversi Skema DMS. Untuk mengedit kode SQL database sumber Anda, gunakan editor SQL biasa Anda. Pilih Segarkan dari database untuk menambahkan kode yang diperbarui dalam proyek migrasi Anda.
- Aturan migrasi dalam Konversi Skema DMS tidak mendukung perubahan pemeriksaan kolom. Selain itu, Anda tidak dapat menggunakan aturan migrasi untuk memindahkan objek ke skema baru.
- Anda tidak dapat menerapkan filter ke pohon basis data sumber dan target Anda untuk menampilkan hanya objek database yang memenuhi klausa filter.

- Paket ekstensi Konversi Skema DMS tidak menyertakan AWS Lambda fungsi yang meniru pengiriman email, penjadwalan pekerjaan, dan fitur lainnya dalam kode konversi Anda.
- Konversi Skema DMS tidak menggunakan kunci KMS yang dikelola pelanggan untuk akses ke sumber daya pelanggan apa pun. AWS Misalnya, Konversi Skema DMS tidak mendukung penggunaan kunci KMS yang dikelola pelanggan untuk mengakses data pelanggan di Amazon S3.

Memulai dengan Konversi Skema DMS

Untuk memulai dengan DMS Schema Conversion, gunakan tutorial berikut. Di dalamnya, Anda dapat belajar mengatur Konversi Skema DMS, membuat proyek migrasi, dan terhubung ke penyedia data Anda. Kemudian, Anda dapat belajar menilai kompleksitas migrasi Anda, dan mengonversi basis data sumber Anda ke format yang kompatibel dengan basis data target Anda. Selain itu, Anda dapat belajar menerapkan kode yang dikonversi ke database target Anda.

Tutorial berikut mencakup tugas-tugas prasyarat dan menunjukkan konversi database Amazon RDS for SQL Server ke Amazon RDS for MySQL. Anda dapat menggunakan salah satu sumber yang didukung dan penyedia data target. Untuk informasi selengkapnya, lihat [Penyedia data sumber untuk Konversi Skema DMS](#).

[Untuk informasi selengkapnya tentang Konversi Skema DMS, baca panduan step-by-step migrasi untuk migrasi Oracle ke PostgreSQL dan SQL Server ke MySQL.](#)

[Video ini](#) memperkenalkan antarmuka pengguna Konversi Skema DMS dan membantu Anda membiasakan diri dengan komponen inti dari layanan ini.

Topik

- [Prasyarat untuk bekerja dengan Konversi Skema DMS](#)
- [Langkah 1: Buat profil instance](#)
- [Langkah 2: Konfigurasikan penyedia data Anda](#)
- [Langkah 3: Buat proyek migrasi](#)
- [Langkah 4: Buat laporan penilaian](#)
- [Langkah 5: Konversi kode sumber Anda](#)
- [Langkah 6: Terapkan kode yang dikonversi](#)
- [Langkah 7: Bersihkan dan pecahkan masalah](#)

Prasyarat untuk bekerja dengan Konversi Skema DMS

Untuk mengatur Konversi Skema DMS, selesaikan tugas-tugas berikut. Kemudian Anda dapat mengatur profil instance, menambahkan penyedia data, dan membuat proyek migrasi.

Topik

- [Buat VPC berdasarkan Amazon VPC](#)
- [Buat bucket Amazon S3.](#)
- [Simpan kredensi basis data di AWS Secrets Manager](#)
- [Buat peran IAM](#)

Buat VPC berdasarkan Amazon VPC

Pada langkah ini, Anda membuat virtual private cloud (VPC) di komputer Anda. Akun AWS VPC ini didasarkan pada layanan Amazon Virtual Private Cloud (Amazon VPC) dan berisi sumber daya Anda. AWS

Untuk membuat VPC untuk Konversi Skema DMS

1. [Masuk ke AWS Management Console dan buka konsol VPC Amazon di https://console.aws.amazon.com/vpc/.](https://console.aws.amazon.com/vpc/)
2. Pilih Buat VPC.
3. Pada halaman Buat VPC, masukkan pengaturan berikut:
 - Sumber daya untuk membuat - VPC dan banyak lagi
 - Generasi otomatis tag nama - Pilih Buat otomatis dan masukkan nama unik secara global. Misalnya, masukkan **sc-vpc**.
 - Blok CIDR IPv4: – **10.0.1.0/24**
 - Gateway NAT - Dalam 1 AZ
 - Titik akhir VPC – Tidak ada
4. Simpan sisa pengaturan apa adanya, lalu pilih Buat VPC.
5. Pilih Subnet, dan catat ID subnet publik dan pribadi Anda.

Untuk terhubung ke database Amazon RDS Anda, buat grup subnet yang menyertakan subnet publik.

Untuk terhubung ke database lokal, buat grup subnet yang menyertakan subnet pribadi. Untuk informasi selengkapnya, lihat [Langkah 1: Buat profil instance](#).

6. Pilih gateway NAT. Pilih gateway NAT Anda dan catat alamat IP Elastis Anda.

Konfigurasi jaringan Anda untuk memastikan bahwa AWS DMS dapat mengakses basis data lokal sumber Anda dari alamat IP publik gateway NAT ini. Untuk informasi selengkapnya, lihat [Menggunakan koneksi internet ke VPC](#).

Gunakan VPC ini saat Anda membuat profil instans dan menargetkan database di Amazon RDS.

Buat bucket Amazon S3.

Untuk menyimpan informasi dari project migrasi, buat bucket Amazon S3. Konversi Skema DMS menggunakan bucket Amazon S3 ini untuk menyimpan item seperti laporan penilaian, kode SQL yang dikonversi, informasi tentang objek skema database, dan sebagainya.

Untuk membuat bucket Amazon S3 untuk Konversi Skema DMS

1. [Masuk ke AWS Management Console dan buka konsol Amazon S3 di https://console.aws.amazon.com/s3/](https://console.aws.amazon.com/s3/).
2. Pilih Buat bucket.
3. Pada halaman Buat bucket, pilih nama unik global untuk bucket S3 Anda. Misalnya, masukkan **sc-s3-bucket**.
4. Untuk Wilayah AWS, pilih Wilayah Anda.
5. Untuk Bucket Versioning, pilih Aktifkan.
6. Simpan sisa pengaturan apa adanya, lalu pilih Buat ember.

Simpan kredensi basis data di AWS Secrets Manager

Menyimpan sumber dan target database Anda di AWS Secrets Manager Pastikan Anda mereplikasi rahasia ini ke Anda Wilayah AWS. Konversi Skema DMS menggunakan rahasia ini untuk terhubung ke database Anda dalam proyek migrasi.

Untuk menyimpan kredensi database Anda di AWS Secrets Manager

1. Masuk ke AWS Management Console dan buka AWS Secrets Manager konsol di <https://console.aws.amazon.com/secretsmanager/>.

2. Pilih Simpan rahasia baru.
3. Halaman Pilih tipe rahasia terbuka. Untuk jenis Rahasia, pilih jenis kredensial database untuk disimpan:
 - Kredensial untuk database Amazon RDS - Pilih opsi ini untuk menyimpan kredensial untuk database Amazon RDS Anda. Untuk Kredensial, masukkan kredensial untuk database Anda. Untuk Database, pilih database Anda.
 - Kredensial untuk database lain - Pilih opsi ini untuk menyimpan kredensial untuk database Oracle atau SQL Server sumber Anda. Untuk Kredensial, masukkan kredensial untuk database Anda.
 - Jenis rahasia lainnya - Pilih opsi ini untuk menyimpan hanya nama pengguna dan kata sandi untuk terhubung ke database Anda. Pilih Tambahkan baris untuk menambahkan dua pasangan kunci-nilai. Pastikan Anda menggunakan **username** dan **password** untuk nama kunci. Untuk nilai yang terkait dengan kunci ini, masukkan kredensial untuk database Anda.
4. Untuk kunci Enkripsi, pilih AWS KMS kunci yang digunakan Secrets Manager untuk mengenkripsi nilai rahasia. Pilih Berikutnya.
5. Pada halaman Konfigurasi rahasia, masukkan nama Rahasia deskriptif. Misalnya, masukkan **sc-source-secret** atau **sc-target-secret**.
6. Pilih Replikasi rahasia dan kemudian Wilayah AWS pilih Wilayah Anda. Pilih Berikutnya.
7. Pada halaman Konfigurasi rotasi, pilih Berikutnya.
8. Pada halaman Ulasan, tinjau detail rahasia Anda, lalu pilih Store.

Untuk menyimpan kredensi untuk basis data sumber dan target Anda, ulangi langkah-langkah ini.

Buat peran IAM

Buat peran AWS Identity and Access Management (IAM) untuk digunakan dalam proyek migrasi Anda. Konversi Skema DMS menggunakan peran IAM ini untuk mengakses bucket Amazon S3 dan kredensial database yang disimpan di dalamnya. AWS Secrets Manager

Untuk membuat peran IAM yang menyediakan akses ke bucket Amazon S3 Anda

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Peran.
3. Pilih Buat peran.

4. Pada halaman Pilih entitas tepercaya, pilih AWS layanan. Pilih DMS.
5. Pilih Berikutnya. Halaman Tambahkan izin terbuka.
6. Untuk kebijakan Filter, masukkan **S3**. Pilih FullAccessAmazonS3.
7. Pilih Berikutnya. Halaman Nama, ulasan, dan buat terbuka.
8. Untuk nama Peran, masukkan nama deskriptif. Misalnya, masukkan **sc-s3-role**. Pilih Buat peran.
9. Pada halaman Peran, masukkan nama **sc-s3-role** Peran. Pilih sc-s3-role.
10. Pada halaman sc-s3-role, pilih tab Trust relationship. Pilih Edit kebijakan kepercayaan.
11. Pada halaman Edit kebijakan kepercayaan, edit hubungan kepercayaan untuk peran yang menggunakan prinsip `schema-conversion.dms.amazonaws.com` layanan sebagai entitas tepercaya.
12. Pilih Perbarui Kebijakan Kepercayaan.

Untuk membuat peran IAM yang menyediakan akses ke AWS Secrets Manager

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Peran.
3. Pilih Buat peran.
4. Pada halaman Pilih entitas tepercaya, pilih AWS layanan. Pilih DMS.
5. Pilih Berikutnya. Halaman Tambahkan izin terbuka.
6. Untuk kebijakan Filter, masukkan **Secret**. Pilih SecretsManagerReadWrite.
7. Pilih Berikutnya. Halaman Nama, ulasan, dan buat terbuka.
8. Untuk nama Peran, masukkan nama deskriptif. Misalnya, masukkan **sc-secrets-manager-role**. Pilih Buat peran.
9. Pada halaman Peran, masukkan nama **sc-secrets-manager-role** Peran. Pilih sc-secrets-manager-role.
10. Pada sc-secrets-manager-rolehalaman, pilih tab Trust relationship. Pilih Edit kebijakan kepercayaan.
11. Pada halaman Edit kebijakan kepercayaan, edit hubungan kepercayaan untuk peran yang akan digunakan `schema-conversion.dms.amazonaws.com` dan kepala layanan AWS DMS regional Anda sebagai entitas tepercaya. Kepala layanan AWS DMS regional ini memiliki format sebagai berikut.

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

Ganti *nama wilayah* nama Wilayah Anda, seperti. us-east-1

Contoh kode berikut menunjukkan prinsipal untuk us-east-1 Wilayah.

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

Contoh kode berikut menunjukkan kebijakan kepercayaan untuk mengakses konversi AWS DMS skema.

```
{
  "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. Pilih Perbarui Kebijakan Kepercayaan.

Langkah 1: Buat profil instance

Sebelum Anda membuat profil instance, konfigurasi grup subnet untuk profil instans Anda. Untuk informasi selengkapnya tentang membuat grup subnet untuk proyek AWS DMS migrasi Anda, lihat [Membuat grup subnet](#).

Anda dapat membuat profil instance seperti yang dijelaskan dalam prosedur berikut. Di profil instance ini, Anda menentukan pengaturan jaringan dan keamanan untuk proyek Konversi Skema DMS Anda.

Buat profil instans

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Di panel navigasi, pilih Profil instans, lalu pilih Buat profil instance.
3. Untuk Nama, masukkan nama unik untuk profil instans Anda. Misalnya, masukkan **sc-instance**.
4. Untuk jenis Jaringan, pilih IPv4 untuk membuat profil instance yang hanya mendukung pengalamatan IPv4. Untuk membuat profil instance yang mendukung pengalamatan IPv4 dan IPv6, pilih mode Dual-stack.
5. Untuk Virtual Private Cloud (VPC), pilih VPC yang Anda buat di langkah prasyarat.
6. Untuk grup Subnet, pilih grup subnet untuk profil instans Anda. Untuk terhubung ke database Amazon RDS, gunakan grup subnet yang menyertakan subnet publik. Untuk terhubung ke database lokal, gunakan grup subnet yang menyertakan subnet pribadi.
7. Pilih Buat profil contoh.

Untuk membuat proyek migrasi, gunakan profil instance ini.

Langkah 2: Konfigurasi penyedia data Anda

Selanjutnya, Anda membuat penyedia data yang menggambarkan basis data sumber dan target Anda. Untuk setiap penyedia data, Anda menentukan jenis penyimpanan data dan informasi lokasi. Anda tidak menyimpan kredensi database Anda di penyedia data.

Untuk membuat penyedia data untuk sumber database lokal

1. Masuk ke AWS Management Console, dan buka AWS DMS konsol.
2. Di panel navigasi, pilih Penyedia data, lalu pilih Buat penyedia data.
3. Untuk Nama, masukkan nama unik untuk penyedia data sumber Anda. Misalnya, masukkan **sc-source**.
4. Untuk tipe Engine, pilih jenis mesin database untuk penyedia data Anda.
5. Berikan informasi koneksi Anda untuk database sumber. Parameter koneksi tergantung pada mesin basis data sumber Anda. Untuk informasi selengkapnya, lihat [Membuat penyedia data](#).

6. Untuk mode Secure Socket Layer (SSL), pilih jenis penegakan SSL.
7. Pilih Buat penyedia data.

Untuk membuat penyedia data untuk database Amazon RDS target

1. Masuk ke AWS Management Console dan buka AWS DMS konsol.
2. Di panel navigasi, pilih Penyedia data, lalu pilih Buat penyedia data.
3. Untuk Konfigurasi, pilih contoh database RDS.
4. Untuk Database dari RDS, pilih Browse, dan pilih database Anda. Konversi Skema DMS secara otomatis mengambil informasi tentang jenis mesin, nama server, dan port.
5. Untuk Nama, masukkan nama unik untuk penyedia data target Anda. Misalnya, masukkan **sc-target**.
6. Untuk Nama basis data, masukkan nama basis data Anda.
7. Untuk mode Secure Socket Layer (SSL), pilih jenis penegakan SSL.
8. Pilih Buat penyedia data.

Langkah 3: Buat proyek migrasi

Sekarang Anda dapat membuat proyek migrasi. Dalam proyek migrasi, Anda menentukan penyedia data sumber dan target, serta profil instans Anda.

Untuk membuat proyek migrasi

1. Pilih Proyek migrasi, lalu pilih Buat proyek migrasi.
2. Untuk Nama, masukkan nama unik untuk proyek migrasi Anda. Misalnya, masukkan **sc-project**.
3. Untuk profil Instance, pilih **sc-instance**.
4. Untuk Sumber, pilih Jelajahi, lalu pilih **sc-source**.
5. Untuk ID Rahasia, pilih **sc-source-secret**.
6. Untuk IAM role, pilih **sc-secrets-manager-role**.
7. Untuk Target, pilih Browse, lalu pilih **sc-target**.
8. Untuk ID Rahasia, pilih **sc-target-secret**.
9. Untuk IAM role, pilih **schema-conversion-role**.

10. Pilih Buat proyek migrasi.

Langkah 4: Buat laporan penilaian

Untuk menilai kompleksitas migrasi, buat laporan penilaian migrasi database. Laporan ini menyertakan daftar semua objek database yang tidak dapat dikonversi secara otomatis oleh Konversi Skema DMS.

Untuk membuat laporan penilaian

1. Pilih proyek Migrasi, lalu pilih **sc-project**.
2. Pilih Konversi skema, lalu pilih Luncurkan konversi skema.
3. Di panel database sumber, pilih skema database untuk dinilai. Juga, pilih kotak centang untuk nama skema ini.
4. Di panel database sumber, pilih Menilai di menu Tindakan. Kotak dialog Assese muncul.
5. Pilih Nilai di kotak dialog untuk mengonfirmasi pilihan Anda.

Tab Ringkasan menunjukkan jumlah item yang Konversi Skema DMS dapat secara otomatis mengkonversi untuk objek penyimpanan database dan objek kode database.

6. Pilih item Tindakan untuk melihat daftar semua objek database yang Konversi Skema DMS tidak dapat mengonversi secara otomatis. Tinjau tindakan yang disarankan untuk setiap item.
7. Untuk menyimpan salinan laporan penilaian Anda, pilih Ekspor hasil. Selanjutnya, pilih salah satu format berikut: CSV atau PDF. Kotak dialog Ekspor muncul.
8. Pilih Ekspor untuk mengonfirmasi pilihan Anda.
9. Pilih ember S3. Konsol Amazon S3 terbuka.
10. Pilih Unduh untuk menyimpan laporan penilaian Anda.

Langkah 5: Konversi kode sumber Anda

Anda dapat mengonversi skema basis data sumber Anda menggunakan prosedur berikut. Kemudian Anda dapat menyimpan kode yang dikonversi sebagai skrip SQL dalam file teks.

Untuk mengonversi skema database Anda

1. Di panel database sumber, pilih skema database untuk dikonversi. Juga, pilih kotak centang untuk nama skema ini.

2. Di panel database sumber, pilih Konversi di menu Tindakan. Kotak dialog Convert muncul.
3. Pilih Konversi di kotak dialog untuk mengonfirmasi pilihan Anda.
4. Pilih objek database di panel database sumber. Konversi Skema DMS menampilkan kode sumber dan kode yang dikonversi untuk objek ini. Anda dapat mengedit kode SQL yang dikonversi untuk objek database menggunakan fitur Edit SQL. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan kode SQL Anda yang telah dikonversi](#).
5. Di panel database target, pilih skema database yang dikonversi. Juga, pilih kotak centang untuk nama skema ini.
6. Untuk Tindakan, pilih Simpan sebagai SQL. Kotak dialog Simpan muncul.
7. Pilih Simpan sebagai SQL untuk mengonfirmasi pilihan Anda.
8. Pilih ember S3. Konsol Amazon S3 terbuka.
9. Pilih Unduh untuk menyimpan skrip SQL Anda.

Langkah 6: Terapkan kode yang dikonversi

Konversi Skema DMS tidak segera menerapkan kode yang dikonversi ke database target Anda. Untuk memperbarui database target Anda, Anda dapat menggunakan skrip SQL yang Anda buat pada langkah sebelumnya. Atau, gunakan prosedur berikut untuk menerapkan kode yang dikonversi dari Konversi Skema DMS.

Untuk menerapkan kode yang dikonversi

1. Di panel database target, pilih skema database yang dikonversi. Juga, pilih kotak centang untuk nama skema ini.
2. Untuk Tindakan, pilih Terapkan perubahan. Kotak dialog Terapkan perubahan muncul.
3. Pilih Terapkan untuk mengonfirmasi pilihan Anda.

Langkah 7: Bersihkan dan pecahkan masalah

Anda dapat menggunakan Amazon CloudWatch untuk meninjau atau membagikan log Konversi Skema DMS Anda.

Untuk meninjau log Konversi Skema DMS

1. Masuk ke AWS Management Console dan buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.

2. Pilih Log, Grup log.

Nama grup log Konversi Skema DMS Anda dimulai dengan `dms-tasks-sct`. Anda dapat mengurutkan grup log berdasarkan waktu Pembuatan untuk menemukan grup log Konversi Skema DMS.

Selain itu, nama grup log Anda menyertakan Nama Sumber Daya Amazon (ARN) proyek migrasi Anda. Anda dapat melihat ARN proyek Anda di halaman proyek Migrasi di Konversi Skema DMS. Pastikan Anda memilih ARN di Preferensi.

3. Pilih nama grup log Anda, lalu pilih nama aliran log Anda.
4. Untuk Tindakan, pilih Ekspor hasil untuk menyimpan log Konversi Skema DMS Anda.

Setelah Anda menyelesaikan konversi skema Anda di Konversi Skema DMS, bersihkan sumber daya Anda.

Untuk membersihkan sumber daya Konversi Skema DMS Anda

1. Masuk ke AWS Management Console dan buka AWS DMS konsol.
2. Di panel navigasi, pilih Proyek migrasi.
 - a. Pilih **sc-project**.
 - b. Pilih Konversi skema, lalu pilih Tutup konversi skema.
 - c. Pilih Hapus dan konfirmasi pilihan Anda.
3. Di panel navigasi, pilih Profil instans.
 - a. Pilih **sc-instance**.
 - b. Pilih Hapus dan konfirmasi pilihan Anda.
4. Di panel navigasi, pilih Penyedia data.
 - a. Pilih **sc-source** dan **sc-target**.
 - b. Pilih Hapus dan konfirmasi pilihan Anda.

Selain itu, pastikan Anda membersihkan AWS sumber daya lain yang Anda buat, seperti bucket Amazon S3, rahasia database, peran IAM AWS Secrets Manager, dan virtual private cloud (VPC).

Menyiapkan jaringan untuk Konversi Skema DMS

Konversi Skema DMS membuat instance konversi skema di cloud pribadi virtual (VPC) berdasarkan layanan Amazon VPC. Saat membuat profil instans, Anda menentukan VPC yang akan digunakan. Anda dapat menggunakan VPC default untuk akun Anda dan Wilayah AWS, atau Anda dapat membuat VPC baru.

Anda dapat menggunakan konfigurasi jaringan yang berbeda untuk mengatur interaksi untuk database sumber dan target Anda dengan Konversi Skema DMS. Konfigurasi ini bergantung pada lokasi penyedia data sumber dan pengaturan jaringan Anda. Topik berikut memberikan deskripsi konfigurasi jaringan umum.

Topik

- [Menggunakan VPC tunggal untuk penyedia sumber dan target data](#)
- [Menggunakan beberapa VPC untuk penyedia sumber dan target data](#)
- [Menggunakan AWS Direct Connect atau VPN untuk mengkonfigurasi jaringan ke VPC](#)
- [Menggunakan koneksi internet ke VPC](#)
- [Menggunakan lingkungan tanpa gateway Internet](#)

Menggunakan VPC tunggal untuk penyedia sumber dan target data

Konfigurasi jaringan paling sederhana untuk DMS Schema Conversion adalah konfigurasi VPC tunggal. Di sini, penyedia data sumber Anda, profil instans, dan penyedia data target semuanya berada di VPC yang sama. Anda dapat menggunakan konfigurasi ini untuk mengonversi database sumber Anda pada instans Amazon EC2.

Untuk menggunakan konfigurasi ini, pastikan grup keamanan VPC yang digunakan oleh profil instans memiliki akses ke penyedia data. Misalnya, Anda dapat mengizinkan rentang VPC Classless Inter-Domain Routing (CIDR) atau alamat IP Elastic untuk gateway Network Address Translation (NAT) Anda.

Menggunakan beberapa VPC untuk penyedia sumber dan target data

Jika penyedia sumber dan target data Anda berada di VPC yang berbeda, Anda dapat membuat profil instans di salah satu VPC. Anda kemudian dapat menautkan kedua VPC ini dengan menggunakan VPC peering. Anda dapat menggunakan konfigurasi ini untuk mengonversi database sumber Anda pada instans Amazon EC2.

SEBUAH koneksi peering VPC adalah koneksi jaringan antara dua VPC yang mengaktifkan routing menggunakan alamat IP pribadi dari setiap VPC seolah-olah mereka berada di jaringan yang sama. Anda dapat membuat koneksi peering VPC antara VPC Anda sendiri, dengan VPC di VPC lain AWS, atau dengan VPC yang berbeda Wilayah AWS. Untuk informasi lebih lanjut mengenai peering VPC, lihat [peering VPC](#) dalam Panduan Pengguna Amazon VPC.

Untuk mengimplementasikan VPC peering, ikuti petunjuk di [Bekerja dengan koneksi peering VPC](#) di dalam Panduan Pengguna Amazon VPC. Pastikan bahwa tabel rute satu VPC berisi blok CIDR yang lain. Misalnya, VPC A menggunakan tujuan 10.0.0.0/16 dan VPC B menggunakan tujuan 172.31.0.0. Dalam hal ini, tabel rute VPC A harus berisi 172.31.0.0, dan tabel rute VPC B harus berisi 10.0.0.0/16. Untuk informasi lebih rinci, lihat [Perbarui tabel rute Anda untuk koneksi peering VPC](#) di dalam Panduan Peering Amazon VPC.

Menggunakan AWS Direct Connect atau VPN untuk mengkonfigurasi jaringan ke VPC

Jaringan jarak jauh dapat terhubung ke VPC menggunakan beberapa opsi, seperti AWS Direct Connect atau koneksi VPN perangkat lunak atau perangkat keras. Anda dapat menggunakan opsi ini untuk mengintegrasikan layanan di tempat yang ada dengan memperluas jaringan internal ke AWS Cloud. Anda dapat mengintegrasikan layanan di tempat seperti pemantauan, otentikasi, keamanan, data, atau sistem lainnya. Dengan menggunakan ekstensi jaringan jenis ini, Anda dapat menghubungkan layanan di tempat dengan lancar ke sumber daya yang dihosting oleh AWS, seperti VPC. Anda dapat menggunakan konfigurasi ini untuk mengonversi database lokal sumber.

Dalam konfigurasi ini, grup keamanan VPC harus menyertakan aturan perutean yang mengirimkan lalu lintas yang ditujukan untuk rentang CIDR VPC atau alamat IP tertentu ke host. Host ini harus dapat menjembatani lalu lintas dari VPC ke VPN on premise. Dalam kasus ini, host NAT mencakup pengaturan grup keamanan sendiri. Pengaturan ini harus memungkinkan lalu lintas dari rentang VPC CIDR atau grup keamanan Anda ke instans NAT. Untuk informasi lebih lanjut, lihat [Buat koneksi VPN Situs-ke-Situs](#) di dalam AWS Site-to-Site VPN Panduan Pengguna.

Menggunakan koneksi internet ke VPC

Jika Anda tidak menggunakan VPN atau AWS Direct Connect untuk terhubung ke AWS sumber daya, Anda dapat menggunakan koneksi internet. Konfigurasi ini melibatkan subnet pribadi di VPC dengan gateway internet. Gateway berisi penyedia data target dan profil instans. Anda dapat menggunakan konfigurasi ini untuk mengonversi database lokal sumber.

Untuk menambahkan gateway internet ke VPC Anda, lihat [Melampirkan gateway internet](#) di Panduan Pengguna Amazon VPC.

Tabel rute VPC harus menyertakan aturan perutean yang mengirim lalu lintas tidak ditakdirkan untuk VPC secara default ke gateway internet. Dalam konfigurasi ini, koneksi ke penyedia data tampaknya berasal dari alamat IP publik gateway NAT Anda. Untuk informasi lebih lanjut, lihat [Tabel Rute VPC](#) di Panduan Pengguna Amazon VPC.

Menggunakan lingkungan tanpa gateway Internet

Untuk membuat lingkungan untuk konversi skema tanpa menggunakan gateway Internet, lakukan hal berikut.

- Ikuti langkah 1-3 di [Memulai](#) tutorial, dengan perubahan berikut:
 - Pilih subnet pribadi, bukan subnet publik.
 - Selama pembuatan instance, untuk Tetapkan IP publik, pilih Tidak.
- Buka Konsol VPC Amazon.
- Pilih Titik akhir, lalu pilih Buat titik akhir.
- Di dalam Buat titik akhir halaman, lakukan hal berikut:
 - Untuk Kategori layanan, pilih AWS Layanan.
 - Di dalam Layanan daftar, pilih `com.amazonaw.{wilayah}.secretsmanajer`
 - Di dalam VPC bagian, pilih VPC yang Anda buat.
 - Pilih subnet untuk VPC Anda.
 - Pilih grup keamanan untuk VPC Anda.
 - Untuk Kebijakan, tinggalkan Akses penuh dipilih.
- Selesaikan sisa [Memulai](#) tutorial.

Membuat penyedia data sumber dalam Konversi Skema DMS

Anda dapat menggunakan database Microsoft SQL Server, Oracle, atau PostgreSQL sebagai penyedia data sumber dalam proyek migrasi untuk Konversi Skema DMS. Penyedia data sumber Anda dapat berupa mesin yang dikelola sendiri yang berjalan di lokasi atau di instans Amazon EC2.

Pastikan Anda mengonfigurasi jaringan untuk mengizinkan interaksi antara penyedia data sumber Anda dan Konversi Skema DMS. Untuk informasi selengkapnya, lihat [Menyiapkan jaringan untuk Konversi Skema DMS](#).

Topik

- [Menggunakan database Microsoft SQL Server sebagai sumber dalam Konversi Skema DMS](#)
- [Menggunakan database Oracle sebagai sumber dalam Konversi Skema DMS](#)
- [Menggunakan database Oracle Data Warehouse sebagai sumber dalam Konversi Skema DMS](#)
- [Menggunakan database PostgreSQL sebagai sumber dalam Konversi Skema DMS](#)
- [Menggunakan database MySQL sebagai sumber dalam Konversi Skema DMS](#)

Menggunakan database Microsoft SQL Server sebagai sumber dalam Konversi Skema DMS

Anda dapat menggunakan database SQL Server sebagai sumber migrasi dalam Konversi Skema DMS.

Anda dapat menggunakan Konversi Skema DMS untuk mengonversi objek kode database dari SQL Server ke target berikut:

- Aurora MySQL
- Aurora PostgreSQL
- RDS for MySQL
- RDS for PostgreSQL

Untuk informasi tentang versi database SQL Server yang didukung, lihat [Penyedia data sumber untuk Konversi Skema DMS](#).

Untuk informasi selengkapnya tentang menggunakan Konversi Skema DMS dengan database SQL Server sumber, lihat panduan migrasi SQL [Server ke MySQL](#). step-by-step

Hak istimewa untuk Microsoft SQL Server sebagai sumber

Lihat daftar hak istimewa berikut yang diperlukan untuk Microsoft SQL Server sebagai sumber:

- LIHAT DEFINISI

- LIHAT STATUS BASIS DATA

Hak VIEW DEFINITION istimewa memungkinkan pengguna yang memiliki akses publik untuk melihat definisi objek. Konversi Skema DMS menggunakan VIEW DATABASE STATE hak istimewa untuk memeriksa fitur edisi SQL Server Enterprise.

Ulangi hibah untuk setiap database yang skema Anda konversi.

Selain itu, berikan hak istimewa berikut pada master database:

- LIHAT STATUS SERVER
- LIHAT DEFINISI APA PUN

Konversi Skema DMS menggunakan VIEW SERVER STATE hak istimewa untuk mengumpulkan pengaturan dan konfigurasi server. Pastikan Anda memberikan VIEW ANY DEFINITION hak istimewa untuk melihat penyedia data.

Untuk membaca informasi tentang Microsoft Analysis Services, jalankan perintah berikut pada master database.

```
EXEC master..sp_addsrvrolemember @loginame = N'<user_name>', @rolename = N'sysadmin'
```

Pada contoh sebelumnya, ganti *<user_name>* placeholder dengan nama pengguna yang sebelumnya Anda berikan dengan hak istimewa yang diperlukan.

Untuk membaca informasi tentang SQL Server Agent, tambahkan pengguna Anda ke peran SQLAgentUser . Jalankan perintah berikut pada msdb database.

```
EXEC sp_addrolemember <SQLAgentRole>, <user_name>;
```

Pada contoh sebelumnya, ganti *<SQLAgentRole>* placeholder dengan nama peran SQL Server Agent. Kemudian ganti *<user_name>* placeholder dengan nama pengguna yang sebelumnya Anda berikan dengan hak istimewa yang diperlukan. Untuk informasi selengkapnya, lihat [Menambahkan pengguna ke AgentUser peran SQL](#) di Panduan Pengguna Amazon RDS.

Untuk mendeteksi pengiriman log, berikan hak SELECT on dbo.log_shipping_primary_databases istimewa pada msdb database.

Untuk menggunakan pendekatan notifikasi replikasi bahasa definisi data (DDL), berikan RECEIVE ON `<schema_name>.<queue_name>` hak istimewa pada basis data sumber Anda. Dalam contoh ini, ganti `<schema_name>` placeholder dengan nama skema database Anda. Kemudian, ganti `<queue_name>` placeholder dengan nama tabel antrian.

Menggunakan database Oracle sebagai sumber dalam Konversi Skema DMS

Anda dapat menggunakan database Oracle sebagai sumber migrasi dalam Konversi Skema DMS.

Untuk terhubung ke database Oracle Anda, gunakan Oracle System ID (SID). Untuk menemukan SID Oracle, kirimkan kueri berikut ke database Oracle Anda:

```
SELECT sys_context('userenv','instance_name') AS SID FROM dual;
```

Anda dapat menggunakan Konversi Skema DMS untuk mengonversi objek kode database dari Oracle Database ke target berikut:

- Aurora MySQL
- Aurora PostgreSQL
- RDS for MySQL
- RDS for PostgreSQL

Untuk informasi tentang versi database Oracle yang didukung, lihat [Penyedia data sumber untuk Konversi Skema DMS](#).

Untuk informasi selengkapnya tentang penggunaan Konversi Skema DMS dengan database Oracle sumber, lihat panduan migrasi [Oracle ke PostgreSQL](#). step-by-step

Hak istimewa untuk Oracle sebagai sumber

Hak istimewa berikut diperlukan untuk Oracle sebagai sumber:

- MENGHUBUNG
- SELECT_CATALOG_ROLE
- PILIH KAMUS APA SAJA
- PILIH DI SYS.ARGUMENT\$

Menggunakan database Oracle Data Warehouse sebagai sumber dalam Konversi Skema DMS

Anda dapat menggunakan database Oracle Data Warehouse sebagai sumber migrasi di Konversi Skema DMS untuk mengonversi objek kode database dan kode aplikasi ke Amazon Redshift.

Untuk informasi tentang versi database Oracle yang didukung, lihat [Penyedia data sumber untuk Konversi Skema DMS](#). Untuk informasi selengkapnya tentang penggunaan Konversi Skema DMS dengan database Oracle sumber, lihat panduan migrasi [Oracle ke PostgreSQL](#). step-by-step

Hak istimewa untuk menggunakan database Oracle Data Warehouse sebagai sumber

Hak istimewa berikut diperlukan untuk Oracle Data Warehouse sebagai sumber:

- MENGHUBUNG
- SELECT_CATALOG_ROLE
- PILIH KAMUS APA SAJA

Pengaturan konversi Oracle Data Warehouse ke Amazon Redshift

Untuk informasi tentang mengedit setelan Konversi Skema DMS, lihat. [Menentukan pengaturan konversi skema untuk proyek migrasi](#)

Pengaturan konversi Oracle Data Warehouse ke Amazon Redshift mencakup yang berikut:

- Tambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi: Pengaturan ini membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi. DMS menambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode yang dikonversi, pilih Error only. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang dikonversi, pilih Semua pesan.

- Jumlah maksimum tabel untuk kluster Amazon Redshift target: Pengaturan ini menetapkan jumlah maksimum tabel yang dapat diterapkan DMS ke cluster Amazon Redshift target Anda. Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node cluster yang berbeda. Pengaturan ini mendukung nilai-nilai berikut:

- Otomatis: DMS menentukan jumlah tabel yang akan diterapkan ke cluster Amazon Redshift target Anda tergantung pada jenis node.
- Tetapkan nilai: Mengatur jumlah tabel secara manual.

DMS mengonversi semua tabel sumber Anda, meskipun jumlah tabel lebih dari yang dapat disimpan oleh cluster Amazon Redshift Anda. DMS menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat menerapkan kode yang dikonversi, DMS akan menampilkan pesan peringatan. Selain itu, DMS menerapkan tabel ke cluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

Untuk informasi tentang kuota tabel Amazon Redshift, lihat Kuota [dan batas di Amazon](#) Redshift.

- Gunakan tampilan UNION ALL: Pengaturan ini memungkinkan Anda mengatur jumlah maksimum tabel target yang dapat dibuat DMS untuk satu tabel sumber.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru partisi tabel dan membuat kueri berjalan lebih cepat, DMS dapat memigrasikan setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, DMS membuat tampilan yang menyertakan data dari semua tabel target yang dibuatnya.

DMS secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan ke cluster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah maksimum tabel target yang dapat dibuat DMS untuk partisi tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun, ditambah dua tabel untuk partisi NO RANGE dan UNKNOWN.

- Elemen format tipe data yang Anda gunakan dalam kode Oracle mirip dengan string format datetime di Amazon Redshift: Gunakan pengaturan ini untuk mengonversi fungsi pemformatan tipe data seperti TO_CHAR dan TO_DATE dengan elemen format TO_NUMBER datetime yang tidak didukung Amazon Redshift. Secara default, DMS menggunakan fungsi paket ekstensi untuk meniru elemen format yang tidak didukung ini dalam kode yang dikonversi.

Model format datetime di Oracle mencakup lebih banyak elemen daripada string format datetime di Amazon Redshift. Jika kode sumber Anda hanya menyertakan elemen format datetime yang didukung Amazon Redshift, tetapkan nilai ini untuk menghindari fungsi paket ekstensi dalam kode yang dikonversi. Menghindari fungsi ekstensi membuat kode yang dikonversi berjalan lebih cepat.

- Elemen format numerik yang Anda gunakan dalam kode Oracle mirip dengan string format numerik di Amazon Redshift: Gunakan pengaturan ini untuk mengonversi fungsi pemformatan tipe data numerik yang tidak didukung Amazon Redshift. Secara default, DMS menggunakan fungsi paket ekstensi untuk meniru elemen format yang tidak didukung ini dalam kode yang dikonversi.

Model format numerik di Oracle mencakup lebih banyak elemen daripada string format numerik di Amazon Redshift. Jika kode sumber Anda hanya menyertakan elemen format numerik yang didukung Amazon Redshift, tetapkan nilai ini untuk menghindari fungsi paket ekstensi dalam kode yang dikonversi. Menghindari fungsi ekstensi membuat kode yang dikonversi berjalan lebih cepat.

- Gunakan fungsi NVL untuk meniru perilaku fungsi Oracle LEAD dan LAG: Jika kode sumber Anda tidak menggunakan nilai default untuk offset dalam LAG fungsi LEAD dan, DMS dapat meniru fungsi ini dengan fungsi tersebut. NVL Secara default, DMS memunculkan item tindakan untuk setiap penggunaan LEAD dan LAG fungsi. Meniru fungsi-fungsi ini menggunakan NVL membuat kode yang dikonversi berjalan lebih cepat.
- Meniru perilaku kunci primer dan unik: Setel setelan ini untuk menyebabkan DMS meniru perilaku kendala kunci primer dan unik pada kluster Amazon Redshift target. Amazon Redshift tidak memberlakukan batasan kunci utama dan unik, dan menggunakannya hanya untuk tujuan informasi. Jika kode sumber Anda menggunakan kendala kunci primer atau unik, setel setelan ini untuk memastikan bahwa DMS mengemulasi perilakunya.
- Gunakan pengkodean kompresi: Setel pengaturan ini untuk menerapkan pengkodean kompresi ke kolom tabel Amazon Redshift. DMS menetapkan pengkodean kompresi secara otomatis menggunakan algoritma Redshift default. Untuk informasi tentang pengkodean kompresi, lihat [Pengkodean kompresi](#) di Panduan Pengembang Database Amazon Redshift.

Amazon Redshift tidak menerapkan kompresi secara default ke kolom yang didefinisikan sebagai kunci pengurutan dan distribusi. Untuk menerapkan kompresi ke kolom ini, atur Gunakan pengkodean kompresi untuk kolom KEY. Anda hanya dapat memilih opsi ini ketika Anda mengatur Gunakan pengkodean kompresi.

Menggunakan database PostgreSQL sebagai sumber dalam Konversi Skema DMS

Anda dapat menggunakan database PostgreSQL sebagai sumber migrasi dalam Konversi Skema DMS.

Anda dapat menggunakan Konversi Skema DMS untuk mengonversi objek kode database dari database PostgreSQL ke target berikut:

- MySQL
- Aurora MySQL

Hak istimewa yang diperlukan untuk PostgreSQL sebagai sumber adalah sebagai berikut:

- TERHUBUNG PADA DATABASE <database_name>
- PENGGUNAAN PADA SKEMA <database_name>
- PILIH PADA SEMUA TABEL DALAM SKEMA <database_name>
- PILIH PADA SEMUA URUTAN DALAM SKEMA <database_name>

Menggunakan database MySQL sebagai sumber dalam Konversi Skema DMS

Anda dapat menggunakan database MySQL sebagai sumber migrasi dalam Konversi Skema DMS.

Anda dapat menggunakan Konversi Skema DMS untuk mengonversi objek kode database dari MySQL Database ke target berikut:

- PostgreSQL
- Aurora PostgreSQL

Hak istimewa yang diperlukan untuk MySQL sebagai sumber adalah sebagai berikut:

- SELECT ON *.*
- SHOW VIEW ON *.*

MySQL ke pengaturan konversi PostgreSQL

Untuk informasi tentang mengedit setelan Konversi Skema DMS, lihat [Menentukan pengaturan konversi skema untuk proyek migrasi](#)

MySQL ke pengaturan konversi PostgreSQL meliputi yang berikut:

- Komentar dalam kode SQL yang dikonversi: Setel pengaturan ini untuk menambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi.

Nilai yang valid:

- Kesalahan saja
- Kesalahan dan peringatan
- Semua pesan

Membuat penyedia data target dalam Konversi Skema DMS

Anda dapat menggunakan database MySQL dan PostgreSQL sebagai penyedia data target dalam proyek migrasi untuk Konversi Skema DMS. Penyedia data target Anda dapat berupa Amazon EC2, Amazon RDS, atau instans Amazon Aurora.

Topik

- [Menggunakan database MySQL sebagai target dalam Konversi Skema DMS](#)
- [Menggunakan database PostgreSQL sebagai target dalam Konversi Skema DMS](#)
- [Menggunakan kluster Amazon Redshift sebagai target dalam Konversi Skema DMS](#)

Menggunakan database MySQL sebagai target dalam Konversi Skema DMS

Anda dapat menggunakan database MySQL sebagai target migrasi di Konversi Skema DMS.

Untuk informasi tentang basis data target yang didukung, lihat [Penyedia data target untuk Konversi Skema DMS](#).

Hak istimewa untuk MySQL sebagai target

Hak istimewa berikut diperlukan untuk MySQL sebagai target:

- 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.*

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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';
```

Pada contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti *your_password* dengan kata sandi yang aman.

Untuk menggunakan Amazon RDS for MySQL atau Aurora MySQL sebagai target, tetapkan parameter ke. `lower_case_table_names` 1 Nilai ini berarti bahwa server MySQL menangani pengidentifikasi nama objek seperti tabel, indeks, pemicu, dan database sebagai case insensitive. Jika Anda telah mengaktifkan logging biner dalam instance target Anda, maka atur `log_bin_trust_function_creators` parameternya ke1. Dalam hal ini, Anda tidak perlu menggunakan `DETERMINISTIC`, `READS SQL DATA` atau `NO SQL` karakteristik untuk membuat fungsi yang disimpan. Untuk mengonfigurasi parameter ini, buat grup parameter DB baru atau modifikasi grup parameter DB yang ada.

Menggunakan database PostgreSQL sebagai target dalam Konversi Skema DMS

Anda dapat menggunakan database PostgreSQL sebagai target migrasi dalam Konversi Skema DMS.

Untuk informasi tentang basis data target yang didukung, lihat [Penyedia data target untuk Konversi Skema DMS](#).

Hak istimewa untuk PostgreSQL sebagai target

Untuk menggunakan PostgreSQL sebagai target, Konversi Skema DMS memerlukan hak istimewa.

`CREATE ON DATABASE` Buat pengguna dan berikan pengguna ini hak istimewa ini untuk setiap database yang ingin Anda gunakan dalam proyek migrasi untuk Konversi Skema DMS.

Untuk menggunakan Amazon RDS for PostgreSQL sebagai target, Konversi Skema DMS memerlukan peran tersebut. `rds_superuser`

Untuk menggunakan sinonim publik yang dikonversi, ubah jalur pencarian default database menggunakan perintah berikut.

```
ALTER DATABASE <db_name> SET SEARCH_PATH = "$user", public_synonyms, public;
```

Dalam contoh ini, ganti `<db_name>` placeholder dengan nama database Anda.

Di PostgreSQL, hanya pemilik skema atau `a` yang dapat menjatuhkan skema. `superuser` Pemilik dapat menjatuhkan skema dan semua objek yang disertakan skema ini, bahkan jika pemilik skema tidak memiliki beberapa objeknya.

Ketika Anda menggunakan pengguna yang berbeda untuk mengonversi dan menerapkan skema yang berbeda ke database target Anda, Anda mungkin menemukan pesan kesalahan ketika Konversi Skema DMS tidak dapat menjatuhkan skema. Untuk menghindari pesan kesalahan ini, gunakan `superuser` peran.

Menggunakan klaster Amazon Redshift sebagai target dalam Konversi Skema DMS

Anda dapat menggunakan database Amazon Redshift sebagai target migrasi dalam Konversi Skema DMS. Untuk informasi tentang basis data target yang didukung, lihat [Penyedia data target untuk Konversi Skema DMS](#).

Hak istimewa untuk Amazon Redshift sebagai target

Menggunakan Amazon Redshift sebagai target untuk Konversi Skema DMS memerlukan hak istimewa berikut:

- `CREATE ON DATABASE`: Memungkinkan DMS untuk membuat skema baru dalam database.
- `CREATE ON SCHEMA`: Memungkinkan DMS untuk membuat objek dalam skema database.
- `PEMBERIAN PENGGUNAAN PADA BAHASA`: Memungkinkan DMS untuk membuat fungsi dan prosedur baru dalam database.
- `GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog`: Menyediakan informasi sistem pengguna tentang cluster Amazon Redshift.
- `GRANT SELECT ON pg_class_info`: Menyediakan informasi pengguna tentang gaya distribusi tabel.

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan izin. Ganti nilai contoh dengan nilai Anda.

```
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;
```

Ulangi GRANT CREATE ON SCHEMA operasi untuk setiap skema target di mana Anda akan menerapkan kode yang dikonversi atau memigrasikan data.

Anda dapat menerapkan paket ekstensi pada basis data Amazon Redshift target Anda. Paket ekstensi adalah modul add-on yang mengemulasi fungsi basis data sumber yang diperlukan saat mengonversi objek ke Amazon Redshift. Untuk informasi selengkapnya, lihat [Menggunakan paket ekstensi dalam Konversi Skema DMS](#).

Mengelola proyek migrasi dalam Konversi Skema DMS

Setelah Anda membuat profil instans dan penyedia data yang kompatibel untuk konversi skema, buat proyek migrasi. Untuk informasi selengkapnya, lihat [Membuat proyek migrasi](#).

Untuk menggunakan proyek baru ini di Konversi Skema DMS, pada halaman Proyek Migrasi, pilih proyek Anda dari daftar. Selanjutnya, pada tab Konversi skema, pilih Luncurkan konversi skema.

Peluncuran pertama Konversi Skema DMS memerlukan beberapa pengaturan. AWS Database Migration Service(AWS DMS) memulai contoh konversi skema, yang memakan waktu hingga 15 menit. Proses ini juga membaca metadata dari database sumber dan target. Setelah peluncuran pertama yang sukses, Anda dapat mengakses Konversi Skema DMS lebih cepat.

Amazon menghentikan instance konversi skema yang digunakan proyek migrasi Anda dalam tiga hari setelah Anda menyelesaikan proyek. Anda dapat mengambil skema dan laporan penilaian yang telah dikonversi dari bucket Amazon S3 yang Anda gunakan untuk Konversi Skema DMS.

Menentukan pengaturan proyek migrasi untuk Konversi Skema DMS

Setelah membuat proyek migrasi dan meluncurkan konversi skema, Anda dapat menentukan setelan proyek migrasi. Anda dapat mengubah pengaturan konversi untuk meningkatkan kinerja kode yang dikonversi. Selain itu, Anda dapat menyesuaikan tampilan konversi skema Anda.

Pengaturan konversi bergantung pada platform basis data sumber dan target Anda. Untuk informasi selengkapnya, silakan lihat [Membuat penyedia data sumber](#) dan [Membuat penyedia data target](#).

Untuk menentukan skema dan database apa yang ingin Anda lihat di panel database sumber dan target, gunakan pengaturan tampilan pohon. Anda dapat menyembunyikan skema kosong, database kosong, database sistem, dan database atau skema yang ditentukan pengguna.

Untuk menyembunyikan database dan skema dalam tampilan pohon

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.
3. Pilih proyek migrasi Anda, dan pada tab Konversi skema pilih Luncurkan konversi skema.
4. Pilih Pengaturan. Halaman Pengaturan terbuka.
5. Di bagian Tree view, lakukan hal berikut:
 - Pilih Sembunyikan skema kosong untuk menyembunyikan skema kosong.
 - Pilih Sembunyikan database kosong untuk menyembunyikan database kosong.
 - Untuk database atau skema sistem, pilih database sistem dan skema dengan nama untuk menyembunyikannya.
 - Untuk database atau skema yang ditentukan pengguna, masukkan nama database dan skema yang ditentukan pengguna yang ingin Anda sembunyikan. Pilih Tambahkan. Nama-nama itu tidak peka huruf besar/kecil.

Untuk menambahkan beberapa database atau skema, gunakan koma untuk memisahkan nama mereka. Untuk menambahkan beberapa objek dengan nama yang mirip, gunakan persen (%) sebagai wildcard. Wildcard ini menggantikan sejumlah simbol apa pun dalam database atau nama skema.

Ulangi langkah-langkah ini untuk bagian Sumber dan Target.

6. Pilih Terapkan, lalu pilih Konversi skema.

Membuat laporan penilaian migrasi database dengan Konversi Skema DMS

Bagian penting dari Konversi Skema DMS adalah laporan yang dihasilkannya untuk membantu Anda mengonversi skema Anda. Laporan penilaian migrasi database ini merangkum semua tugas konversi skema. Ini juga merinci item tindakan untuk skema yang tidak dapat dikonversi ke mesin DB instans

DB target Anda. Anda dapat melihat laporan di AWS DMS konsol atau menyimpan salinan laporan ini sebagai file PDF atau nilai dipisahkan koma (CSV).

Laporan penilaian migrasi mencakup hal-hal berikut:

- Ringkasan eksekutif
- Rekomendasi, termasuk konversi objek server, saran cadangan, dan perubahan server terkait

Bila Anda memiliki item yang tidak dapat dikonversi oleh Konversi Skema DMS secara otomatis, laporan tersebut memberikan perkiraan berapa banyak upaya yang diperlukan untuk menulis kode yang setara untuk instans DB target Anda.

Topik

- [Membuat laporan penilaian migrasi basis data](#)
- [Melihat laporan penilaian migrasi basis data](#)
- [Menyimpan laporan penilaian migrasi database](#)

Membuat laporan penilaian migrasi basis data

Setelah membuat proyek migrasi, gunakan prosedur berikut untuk membuat laporan penilaian migrasi database.

Membuat laporan penilaian migrasi database

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih Proyek migrasi. Halaman proyek Migrasi terbuka.
3. Pilih proyek migrasi Anda, lalu pilih Konversi skema.
4. Pilih Luncurkan konversi skema. Halaman konversi Skema terbuka.
5. Di panel database sumber, pilih skema database atau item skema yang ingin Anda nilai. Untuk menyertakan beberapa objek dalam laporan, pastikan Anda memilih semua item.
6. Setelah Anda memilih kotak centang untuk semua objek skema yang ingin Anda nilai, Anda harus memilih node induk untuk objek yang dipilih. Menu Tindakan di panel database sumber sekarang tersedia.
7. Pilih Nilai di menu Tindakan. Sebuah kotak dialog konfirmasi kemudian muncul.
8. Pilih Nilai di kotak dialog untuk mengonfirmasi pilihan Anda.

Melihat laporan penilaian migrasi basis data

Setelah Anda membuat laporan penilaian, Konversi Skema DMS menambahkan informasi di tab berikut:

- Ringkasan
- Item tindakan

Itu Ringkasan tab menunjukkan jumlah item yang dapat dikonversi secara otomatis oleh Konversi Skema DMS.

Tab Item tindakan menunjukkan item yang tidak dapat dikonversi oleh Konversi Skema DMS secara otomatis, dan memberikan rekomendasi tentang cara mengelola item ini.

Laporan penilaian

Tab Ringkasan menampilkan informasi ringkasan dari laporan penilaian migrasi database. Ini menunjukkan jumlah item yang DMS Schema Konversi dapat secara otomatis mengkonversi untuk objek penyimpanan database dan objek kode database.

Dalam kebanyakan kasus, Konversi Skema DMS tidak dapat secara otomatis mengubah semua item skema ke mesin database target. Tab Ringkasan memberikan perkiraan upaya yang diperlukan untuk membuat item skema dalam instans DB target Anda yang setara dengan yang ada di sumber Anda.

Untuk melihat ringkasan konversi untuk objek penyimpanan database seperti tabel, urutan, kendala, tipe data, dan sebagainya, pilih objek penyimpanan database.

Untuk melihat ringkasan konversi untuk objek kode database seperti prosedur, fungsi, tampilan, pemicu, dan sebagainya, pilih objek kode database.

Untuk mengubah ruang lingkup laporan penilaian, pilih node yang diperlukan di pohon basis data sumber. Konversi Skema DMS memperbarui ringkasan laporan penilaian agar sesuai dengan lingkup yang dipilih.

Item tindakan laporan penilaian

Itu Item tindakan tab berisi daftar item yang Konversi Skema DMS tidak dapat secara otomatis dikonversi ke format yang kompatibel dengan mesin database target. Untuk setiap item tindakan,

Konversi Skema DMS memberikan deskripsi masalah dan tindakan yang disarankan. Konversi Skema DMS mengelompokkan item tindakan serupa dan menampilkan jumlah kejadian.

Untuk melihat kode untuk objek database terkait, pilih item tindakan dalam daftar.

Menyimpan laporan penilaian migrasi database

Setelah membuat laporan penilaian migrasi database, Anda dapat menyimpan salinan laporan ini sebagai file PDF atau nilai dipisahkan koma (CSV).

Untuk menyimpan laporan penilaian migrasi database sebagai file PDF

1. Pilih Ekspor, lalu pilih PDF. Tinjau kotak dialog, dan pilih Ekspor ke PDF.
2. Konversi Skema DMS membuat arsip dengan file PDF Anda dan menyimpan arsip ini di bucket Amazon S3 Anda. Untuk mengubah bucket Amazon S3, edit pengaturan konversi skema di profil instans Anda.
3. Buka file laporan penilaian di bucket Amazon S3.

Untuk menyimpan laporan penilaian migrasi database sebagai file CSV

1. Pilih Ekspor, lalu pilih CSV. Tinjau kotak dialog, dan pilih Ekspor ke CSV.
2. Konversi Skema DMS membuat arsip dengan file CSV dan menyimpan arsip ini di bucket Amazon S3 Anda. Untuk mengubah bucket Amazon S3, edit pengaturan konversi skema di profil instans Anda.
3. Buka file laporan penilaian di bucket Amazon S3.

File PDF berisi informasi ringkasan dan item tindakan.

Saat Anda mengekspor laporan penilaian ke CSV, Konversi Skema DMS membuat tiga file CSV.

File CSV pertama berisi informasi berikut tentang item CSV pertama berisi informasi berikut tentang item CSV pertama berisi informasi berikut tentang item CSV berikut tentang

- Kategori
- Kejadian
- Item tindakan
- Subjek

- Grup
- Deskripsi
- Referensi dokumentasi
- Tindakan yang direkomendasikan
- Baris
- Posisi
- Sumber
- Target
- Alamat IP server dan port
- Basis Data
- Skema

File CSV kedua menyertakan `Action_Items_Summary` akhiran dalam namanya dan berisi informasi berikut:

- Skema
- Item tindakan
- Jumlah kejadian
- Upaya kurva pembelajaran, yang merupakan jumlah upaya yang diperlukan untuk merancang pendekatan untuk mengubah setiap item tindakan
- Upaya untuk mengonversi kejadian item tindakan, yang menunjukkan upaya yang diperlukan untuk mengonversi setiap item tindakan, mengikuti pendekatan yang dirancang
- Deskripsi item tindakan
- Tindakan yang direkomendasikan

Nilai-nilai yang menunjukkan tingkat upaya yang diperlukan didasarkan pada skala tertimbang, mulai dari rendah (paling sedikit) hingga tinggi (paling).

File CSV ketiga termasuk `Summary` dalam namanya dan berisi informasi berikut:

- Kategori
- Jumlah objek
- Objek otomatis dikonversi

- Objek dengan tindakan sederhana
- Objek dengan tindakan kompleksitas sedang
- Objek dengan tindakan kompleks
- Total baris kode

Menggunakan Konversi Skema DMS

Konversi Skema DMS mengonversi skema database yang ada dan sebagian besar objek kode database ke format yang kompatibel dengan database target.

Konversi Skema DMS mengotomatiskan sebagian besar proses konversi skema database pemrosesan transaksi online (OLTP) Anda ke Amazon RDS untuk MySQL atau RDS untuk PostgreSQL. Mesin basis data sumber dan target berisi banyak fitur dan kemampuan yang berbeda, dan Konversi Skema DMS mencoba untuk membuat skema yang setara sedapat mungkin. Untuk objek database di mana konversi langsung tidak memungkinkan, DMS Schema Conversion menyediakan daftar tindakan untuk Anda ambil.

Untuk mengonversi skema database Anda, gunakan proses berikut:

- Sebelum Anda mengonversi skema database Anda, atur aturan transformasi yang mengubah nama objek database Anda selama konversi.
- Buat laporan penilaian migrasi database untuk memperkirakan kompleksitas migrasi. Laporan ini memberikan rincian tentang elemen skema yang Konversi Skema DMS tidak dapat mengonversi secara otomatis.
- Konversikan penyimpanan basis data sumber dan objek kode Anda. Konversi Skema DMS membuat versi lokal dari objek database yang dikonversi. Anda dapat mengakses objek yang dikonversi ini dalam proyek migrasi Anda.
- Simpan kode yang dikonversi ke file SQL untuk meninjau, mengedit, atau menangani item tindakan konversi. Secara opsional, terapkan kode yang dikonversi langsung ke database target Anda.

Untuk mengonversi skema gudang data, gunakan desktop AWS Schema Conversion Tool. Untuk informasi selengkapnya, lihat [Mengonversi skema gudang data ke Amazon Redshift](#) di Panduan Pengguna Alat AWS Konversi Skema.

Topik

- [Menyiapkan aturan transformasi dalam Konversi Skema DMS](#)

- [Mengonversi skema database dalam Konversi Skema DMS](#)
- [Menentukan pengaturan konversi skema untuk proyek migrasi](#)
- [Menyegarkan skema database Anda dalam Konversi Skema DMS](#)
- [Menyimpan dan menerapkan kode konversi Anda dalam Konversi Skema DMS](#)

Menyiapkan aturan transformasi dalam Konversi Skema DMS

Sebelum Anda mengonversi skema database Anda dengan Konversi Skema DMS, Anda dapat mengatur aturan transformasi. Aturan transformasi dapat melakukan hal-hal seperti mengubah nama objek menjadi huruf kecil atau huruf besar, menambah atau menghapus awalan atau akhiran, dan mengganti nama objek. Misalnya, anggaplah Anda memiliki satu set tabel dalam skema sumber Anda bernama `test_TABLE_NAME`. Anda dapat mengatur aturan yang mengubah awalan `test_` ke awalan `demo_` dalam skema target.

Anda dapat membuat aturan transformasi yang melakukan tugas-tugas berikut:

- Menambahkan, menghapus, atau mengganti awalan
- Menambahkan, menghapus, atau mengganti sufiks
- Mengubah tipe data kolom
- Ubah nama objek menjadi huruf kecil atau huruf besar
- Ganti nama objek

Anda dapat membuat aturan transformasi untuk objek berikut:

- Skema
- Tabel
- Kolom

Membuat aturan transformasi

Konversi Skema DMS menyimpan aturan transformasi sebagai bagian dari proyek migrasi Anda. Anda dapat mengatur aturan transformasi saat membuat proyek migrasi, atau mengeditnya nanti.

Anda dapat menambahkan beberapa aturan transformasi dalam proyek Anda. Konversi Skema DMS menerapkan aturan transformasi selama konversi dalam urutan yang sama seperti yang Anda tambahkan.

Untuk membuat aturan transformasi

1. Pada halaman Buat proyek migrasi, pilih Tambahkan aturan transformasi. Untuk informasi selengkapnya, lihat [Membuat proyek migrasi](#).
2. Untuk target Aturan, pilih jenis objek database yang aturan ini berlaku.
3. Untuk skema Sumber, pilih Masukkan skema. Kemudian, masukkan nama skema sumber, tabel, dan kolom tempat aturan ini berlaku. Anda dapat memasukkan nama yang tepat untuk memilih satu objek, atau Anda dapat memasukkan pola untuk memilih beberapa objek. Gunakan persen (%) sebagai wildcard untuk mengganti sejumlah simbol apa pun dalam nama objek database.
4. Untuk Tindakan, pilih tugas yang akan dilakukan.
5. Tergantung pada jenis aturan, masukkan satu atau dua nilai tambahan. Misalnya, untuk mengganti nama objek, masukkan nama baru objek. Untuk mengganti awalan, masukkan awalan lama dan awalan baru.
6. Pilih Tambahkan aturan transformasi untuk menambahkan aturan transformasi lain.

Setelah selesai menambahkan aturan, pilih Buat proyek migrasi.

Untuk menduplikasi aturan transformasi yang ada, pilih Duplikat. Untuk mengedit aturan transformasi yang ada, pilih aturan dari daftar. Untuk menghapus aturan transformasi yang ada, pilih Hapus.

Mengedit aturan transformasi

Anda dapat menambahkan baru, menghapus, atau mengedit aturan transformasi yang ada di proyek migrasi Anda. Karena Konversi Skema DMS menerapkan aturan transformasi selama peluncuran konversi skema, pastikan Anda menutup konversi skema dan meluncurkannya lagi setelah Anda mengedit aturan.

Untuk mengedit aturan transformasi

1. Masuk ke AWS Management Console, dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih Proyek migrasi, lalu pilih proyek migrasi Anda.
3. Pilih Konversi skema, lalu pilih Tutup konversi skema.
4. Setelah AWS DMS menutup konversi skema, pilih Ubah untuk mengedit pengaturan proyek migrasi Anda.
5. Untuk aturan Transformasi, pilih salah satu tindakan berikut:

- Pilih Duplikat untuk menduplikasi aturan transformasi yang ada dan menambahkannya di akhir daftar.
 - Pilih Hapus untuk menghapus aturan transformasi yang ada.
 - Pilih aturan transformasi yang ada untuk mengeditnya.
6. Setelah Anda selesai mengedit aturan, pilih Simpan perubahan.
 7. Pada halaman proyek Migrasi, pilih proyek Anda dari daftar. Pilih Konversi skema, lalu pilih Luncurkan konversi skema.

Mengonversi skema database dalam Konversi Skema DMS

Setelah membuat proyek migrasi dan terhubung ke basis data sumber dan target, Anda dapat mengonversi objek basis data sumber Anda ke format yang kompatibel dengan basis data target Anda. Konversi Skema DMS menampilkan skema basis data sumber Anda di panel kiri dalam format tampilan pohon.

Setiap node dari pohon database malas dimuat. Ketika Anda memilih node dalam tampilan pohon, DMS Schema Conversion meminta informasi skema dari database sumber Anda pada saat itu. Untuk memuat informasi skema lebih cepat, pilih skema Anda, lalu pilih Muat metadata dari menu Tindakan. Konversi Skema DMS kemudian membaca metadata database dan menyimpan informasi di bucket Amazon S3. Anda sekarang dapat menelusuri objek database lebih cepat.

Anda dapat mengonversi seluruh skema database, atau Anda dapat memilih item skema apa pun dari database sumber Anda untuk dikonversi. Jika item skema yang Anda pilih bergantung pada item induk, maka Konversi Skema DMS juga menghasilkan skema untuk item induk. Misalnya, saat Anda memilih tabel untuk dikonversi, Konversi Skema DMS membuat tabel yang dikonversi dan skema database tempat tabel berada.

Mengonversi objek database

Anda dapat menggunakan Konversi Skema DMS untuk mengonversi seluruh skema database atau objek skema database terpisah.

Untuk mengonversi seluruh skema basis data

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.

3. Pilih proyek migrasi Anda, lalu pilih Konversi skema.
4. Pilih Luncurkan konversi skema. Halaman konversi Skema terbuka.
5. Di panel database sumber, pilih kotak centang untuk nama skema.
6. Pilih skema ini di panel kiri proyek migrasi. Konversi Skema DMS menyoroti nama skema dengan warna biru dan mengaktifkan menu Tindakan.
7. Untuk Tindakan, pilih Konversi. Kotak dialog konversi muncul.
8. Pilih Konversi di kotak dialog untuk mengonfirmasi pilihan Anda.

Untuk mengonversi objek basis data sumber Anda

1. Masuk ke AWS Management Console, dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.
3. Pilih proyek migrasi Anda, lalu pilih Konversi skema.
4. Pilih Luncurkan konversi skema. Halaman konversi Skema terbuka.
5. Di panel database sumber, pilih objek database sumber Anda.
6. Setelah Anda memilih semua kotak centang untuk objek yang ingin Anda konversi, pilih node induk untuk semua objek yang dipilih di panel kiri Anda.

Konversi Skema DMS menyoroti node induk dengan warna biru dan mengaktifkan menu Tindakan.

7. Untuk Tindakan, pilih Konversi. Kotak dialog konversi muncul.
8. Pilih Konversi di kotak dialog untuk mengonfirmasi pilihan Anda.

Misalnya, untuk mengonversi dua dari 10 tabel, pilih kotak centang untuk dua tabel yang ingin Anda konversi. Perhatikan bahwa menu Tindakan tidak aktif. Setelah Anda memilih node Tables, DMS Schema Conversion menyoroti namanya dengan warna biru dan mengaktifkan menu Actions. Kemudian Anda dapat memilih Konversi dari menu ini.

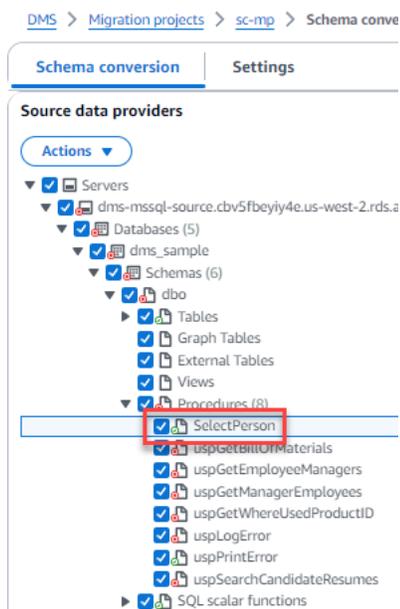
Demikian juga, untuk mengonversi dua tabel dan tiga prosedur, pilih kotak centang untuk nama objek. Kemudian, pilih simpul skema untuk mengaktifkan menu Tindakan, dan pilih Konversi skema.

Mengedit dan menyimpan kode SQL Anda yang telah dikonversi

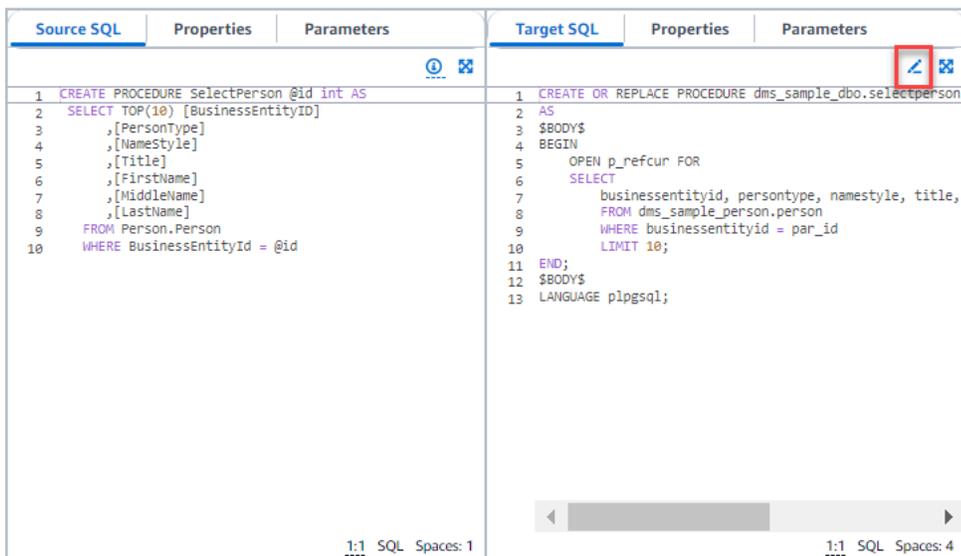
Halaman konversi Skema memungkinkan Anda untuk mengedit kode SQL dikonversi dalam objek database Anda. Gunakan prosedur berikut untuk mengedit kode SQL Anda yang telah dikonversi, menerapkan perubahan, dan kemudian menyimpannya.

Untuk mengedit, menerapkan perubahan ke, dan menyimpan kode SQL Anda yang telah dikonversi

1. Di halaman konversi Skema, buka tampilan pohon di panel Penyedia data sumber untuk menampilkan objek kode.



2. Dari panel Penyedia data sumber, pilih Tindakan, Konversi. Konfirmasikan tindakan.
3. Saat konversi selesai, untuk melihat SQL yang dikonversi, perluas panel tengah jika diperlukan. Untuk mengedit SQL yang dikonversi, pilih ikon edit di panel Target SQL.



4. Setelah Anda mengedit SQL target, konfirmasi perubahan Anda dengan memilih ikon centang di bagian atas halaman. Konfirmasikan tindakan.
5. Di panel Penyedia data target, pilih Tindakan, Terapkan perubahan. Konfirmasikan tindakan.
6. DMS menulis prosedur yang diedit ke penyimpanan data target.

Meninjau objek database yang dikonversi

Setelah Anda mengonversi objek database sumber Anda, Anda dapat memilih objek di panel kiri proyek Anda. Anda kemudian dapat melihat sumber dan kode yang dikonversi untuk objek itu. Konversi Skema DMS secara otomatis memuat kode yang dikonversi untuk objek yang Anda pilih di panel kiri. Anda juga dapat melihat properti atau parameter objek yang Anda pilih.

Konversi Skema DMS secara otomatis menyimpan kode yang dikonversi sebagai bagian dari proyek migrasi Anda. Itu tidak menerapkan perubahan kode ini ke database target Anda. Untuk informasi selengkapnya tentang menerapkan kode yang dikonversi ke database target Anda, lihat [Menerapkan kode konversi Anda](#). Untuk menghapus kode yang dikonversi dari proyek migrasi, pilih skema target di panel kanan, lalu pilih Segarkan dari database dari Tindakan.

Setelah mengonversi objek basis data sumber, Anda dapat melihat ringkasan konversi dan item tindakan di panel tengah bawah. Anda dapat melihat informasi yang sama saat membuat laporan penilaian. Laporan penilaian berguna untuk mengidentifikasi dan menyelesaikan item skema yang tidak dapat dikonversi oleh Konversi Skema DMS. Anda dapat menyimpan ringkasan laporan penilaian dan daftar item tindakan konversi dalam file CSV. Untuk informasi selengkapnya, lihat [Laporan penilaian migrasi database](#).

Menentukan pengaturan konversi skema untuk proyek migrasi

Setelah membuat proyek migrasi, Anda dapat menentukan pengaturan konversi di Konversi Skema DMS. Mengonfigurasi pengaturan konversi skema Anda meningkatkan kinerja kode yang dikonversi.

Untuk mengedit setelah konversi

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.
3. Pilih proyek migrasi Anda. Pilih Konversi skema, lalu Luncurkan konversi skema.
4. Pilih Pengaturan. Halaman Pengaturan terbuka.
5. Di bagian Konversi, ubah pengaturan.
6. Pilih Terapkan, lalu pilih Konversi skema.

Untuk semua pasangan konversi, Anda dapat membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi. Untuk membatasi jumlah komentar dalam kode yang dikonversi, buka pengaturan konversi di proyek migrasi Anda.

Untuk Komentar dalam kode SQL yang dikonversi, pilih tingkat keparahan item tindakan. Konversi Skema DMS menambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi. Misalnya, untuk meminimalkan jumlah komentar dalam kode yang dikonversi, pilih Error only.

Untuk menyertakan komentar untuk semua item tindakan dalam kode yang dikonversi, pilih Semua pesan.

Pengaturan konversi lainnya berbeda untuk setiap pasangan basis data sumber dan target.

Topik

- [Pengaturan konversi Oracle ke MySQL](#)
- [Pengaturan konversi Oracle ke PostgreSQL](#)
- [SQL Server ke pengaturan konversi MySQL](#)
- [SQL Server ke pengaturan konversi PostgreSQL](#)
- [PostgreSQL ke pengaturan konversi MySQL](#)

Pengaturan konversi Oracle ke MySQL

Oracle ke pengaturan konversi MySQL di DMS Schema Conversion meliputi:

- Database Oracle sumber Anda dapat menggunakan ROWID pseudocolumn. MySQL tidak mendukung fungsionalitas serupa. Konversi Skema DMS dapat meniru ROWID pseudocolumn dalam kode yang dikonversi. Untuk melakukannya, aktifkan opsi Generate row ID.

Jika kode Oracle sumber Anda tidak menggunakan ROWID pseudocolumn, matikan opsi Generate row ID. Dalam hal ini, kode yang dikonversi bekerja lebih cepat.

- Kode Oracle sumber Anda dapat menyertakan `TO_CHAR`, `TO_DATE`, dan `TO_NUMBER` fungsi dengan parameter yang tidak didukung MySQL. Secara default, Konversi Skema DMS mengemulasi penggunaan parameter ini dalam kode yang dikonversi.

Anda dapat menggunakan `TO_CHAR` MySQL asli `TO_DATE`, `TO_NUMBER` dan fungsi ketika kode Oracle sumber Anda tidak memiliki parameter yang tidak didukung oleh MySQL. Dalam hal ini, kode yang dikonversi bekerja lebih cepat. Untuk melakukannya, pilih nilai berikut:

- Gunakan fungsi MySQL `TO_CHAR` asli
- Gunakan fungsi asli MySQL `TO_DATE`
- Gunakan fungsi MySQL `TO_NUMBER` asli
- Database dan aplikasi Anda dapat berjalan di zona waktu yang berbeda. Secara default, Konversi Skema DMS mengemulasi zona waktu dalam kode yang dikonversi. Namun, Anda tidak memerlukan emulasi ini ketika database dan aplikasi Anda menggunakan zona waktu yang sama. Dalam hal ini, pilih Tingkatkan kinerja kode yang dikonversi di mana database dan aplikasi menggunakan zona waktu yang sama.

Pengaturan konversi Oracle ke PostgreSQL

Pengaturan konversi Oracle ke PostgreSQL dalam Konversi Skema DMS meliputi:

- AWS DMS dapat mengonversi tampilan terwujud Oracle ke tabel atau tampilan terwujud di PostgreSQL. Untuk tampilan Terwujud, pilih cara mengonversi tampilan terwujud sumber Anda.
- Database Oracle sumber Anda dapat menggunakan ROWID pseudocolumn. PostgreSQL tidak mendukung fungsionalitas serupa. Konversi Skema DMS dapat meniru ROWID pseudocolumn dalam kode yang dikonversi menggunakan tipe data atau `bigint character varying` Untuk melakukannya, pilih Gunakan tipe data `bigint` untuk meniru pseudocolumn ROWID atau Gunakan karakter yang bervariasi tipe data untuk meniru pseudocolumn ROWID untuk Row ID.

Jika kode Oracle sumber Anda tidak menggunakan ROWID pseudocolumn, pilih Jangan buat. Dalam hal ini, kode yang dikonversi bekerja lebih cepat.

- Kode Oracle sumber Anda dapat menyertakan TO_CHAR, TO_DATE, dan TO_NUMBER fungsi dengan parameter yang tidak didukung PostgreSQL. Secara default, Konversi Skema DMS mengemulasi penggunaan parameter ini dalam kode yang dikonversi.

Anda dapat menggunakan TO_CHAR PostgreSQL asli TO_DATE,, TO_NUMBER dan fungsi ketika kode Oracle sumber Anda tidak memiliki parameter yang tidak didukung oleh PostgreSQL. Dalam hal ini, kode yang dikonversi bekerja lebih cepat. Untuk melakukannya, pilih nilai berikut:

- Gunakan fungsi PostgreSQL TO_CHAR asli
- Gunakan fungsi PostgreSQL TO_DATE asli
- Gunakan fungsi PostgreSQL TO_NUMBER asli
- Database dan aplikasi Anda dapat berjalan di zona waktu yang berbeda. Secara default, Konversi Skema DMS mengemulasi zona waktu dalam kode yang dikonversi. Namun, Anda tidak memerlukan emulasi ini ketika database dan aplikasi Anda menggunakan zona waktu yang sama. Dalam hal ini, pilih Tingkatkan kinerja kode yang dikonversi di mana database dan aplikasi menggunakan zona waktu yang sama.
- Untuk terus menggunakan urutan dalam kode yang dikonversi, pilih Isi urutan yang dikonversi dengan nilai terakhir yang dihasilkan di sisi sumber.
- Dalam beberapa kasus, database Oracle sumber Anda mungkin hanya menyimpan nilai integer di kolom kunci primer atau asing dari tipe NUMBER data. Dalam kasus ini, AWS DMS dapat mengkonversi kolom ini ke tipe BIGINT data. Pendekatan ini meningkatkan kinerja kode Anda yang dikonversi. Untuk melakukannya, pilih Mengkonversi kolom kunci primer dan asing dari tipe data NUMBER ke tipe data BIGINT. Pastikan sumber Anda tidak menyertakan nilai floating point di kolom ini untuk menghindari kehilangan data.
- Untuk melewati pemicu dan kendala yang dinonaktifkan dalam kode sumber Anda, pilih Konversi hanya pemicu dan kendala aktif.
- Anda dapat menggunakan DMS Schema Conversion untuk mengkonversi variabel string yang disebut sebagai SQL dinamis. Kode database Anda dapat mengubah nilai variabel string ini. Untuk memastikan bahwa AWS DMS selalu mengonversi nilai terbaru dari variabel string ini, pilih Konversi kode SQL dinamis yang dibuat dalam rutinitas yang disebut.
- PostgreSQL versi 10 dan sebelumnya tidak mendukung prosedur. Jika Anda tidak terbiasa menggunakan prosedur di PostgreSQL, dapat mengonversi prosedur Oracle ke fungsi PostgreSQL AWS DMS . Untuk melakukannya, pilih Konversi prosedur ke fungsi.

- Untuk melihat informasi tambahan tentang item tindakan yang terjadi, Anda dapat menambahkan fungsi tertentu ke paket ekstensi. Untuk melakukannya, pilih Tambahkan fungsi paket ekstensi yang meningkatkan pengecualian yang ditentukan pengguna. Kemudian pilih tingkat keparahan untuk meningkatkan pengecualian yang ditentukan pengguna. Pastikan Anda menerapkan skema paket ekstensi setelah mengonversi objek basis data sumber Anda. Untuk informasi selengkapnya tentang paket ekstensi, lihat [Menggunakan paket ekstensi](#).
- Database Oracle sumber Anda dapat menyertakan batasan dengan nama yang dibuat secara otomatis. Jika kode sumber Anda menggunakan nama-nama ini, pastikan Anda memilih Simpan nama kendala yang dihasilkan sistem. Jika kode sumber Anda menggunakan batasan ini, tetapi tidak menggunakan namanya, hapus opsi ini untuk meningkatkan kecepatan konversi.
- Jika basis data sumber dan target Anda berjalan di zona waktu yang berbeda, fungsi yang mengemulasi fungsi Oracle SYSDATE bawaan mengembalikan nilai yang berbeda dibandingkan dengan fungsi sumber. Untuk memastikan bahwa fungsi sumber dan target Anda mengembalikan nilai yang sama, pilih Tetapkan zona waktu database sumber Anda.
- Anda dapat menggunakan fungsi dari ekstensi orafce dalam kode yang dikonversi. Untuk melakukannya, untuk rutinitas bawaan Orafce, pilih fungsi yang akan digunakan. Untuk informasi lebih lanjut tentang orafce, lihat [orafce](#) di GitHub

SQL Server ke pengaturan konversi MySQL

SQL Server ke MySQL pengaturan konversi di DMS Schema Conversion meliputi:

- Database SQL Server sumber Anda dapat menyimpan output EXEC dalam tabel. Konversi Skema DMS membuat tabel sementara dan prosedur tambahan untuk meniru fitur ini. Untuk menggunakan emulasi ini, pilih Buat rutinitas tambahan untuk menangani kumpulan data terbuka.

SQL Server ke pengaturan konversi PostgreSQL

SQL Server ke pengaturan konversi PostgreSQL dalam Konversi Skema DMS meliputi:

- Di SQL Server, Anda dapat menggunakan indeks dengan nama yang sama di tabel yang berbeda. Namun, di PostgreSQL, semua nama indeks yang Anda gunakan dalam skema harus unik. Untuk memastikan bahwa Konversi Skema DMS menghasilkan nama unik untuk semua indeks Anda, pilih Hasilkan nama unik untuk indeks.

- PostgreSQL versi 10 dan sebelumnya tidak mendukung prosedur. Jika Anda tidak terbiasa menggunakan prosedur di PostgreSQL, dapat mengonversi prosedur SQL Server ke fungsi AWS DMS PostgreSQL. Untuk melakukannya, pilih Konversi prosedur ke fungsi.
- Database SQL Server sumber Anda dapat menyimpan output EXEC dalam tabel. Konversi Skema DMS membuat tabel sementara dan prosedur tambahan untuk meniru fitur ini. Untuk menggunakan emulasi ini, pilih Buat rutinitas tambahan untuk menangani kumpulan data terbuka.
- Anda dapat menentukan template yang akan digunakan untuk nama skema dalam kode yang dikonversi. Untuk nama Skema, pilih salah satu opsi berikut:
 - DB — Menggunakan nama database SQL Server sebagai nama skema di PostgreSQL.
 - SKEMA — Menggunakan nama skema SQL Server sebagai nama skema di PostgreSQL.
 - DB_SCHEMA — Menggunakan kombinasi database SQL Server dan nama skema sebagai nama skema di PostgreSQL.
- Anda dapat menyimpan huruf dari nama objek sumber Anda. Untuk menghindari konversi nama objek ke huruf kecil, pilih Simpan nama objek dalam kasus yang sama. Opsi ini hanya berlaku ketika Anda mengaktifkan opsi sensitivitas kasus di basis data target Anda.
- Anda dapat menyimpan nama parameter dari database sumber Anda. Konversi Skema DMS dapat menambahkan tanda kutip ganda ke nama parameter dalam kode yang dikonversi. Untuk melakukannya, pilih Simpan nama parameter asli.
- Anda dapat menyimpan panjang parameter rutin dari database sumber Anda. Konversi Skema DMS membuat domain dan menggunakannya untuk menentukan panjang parameter rutin. Untuk melakukannya, pilih Pertahankan panjang parameter.

PostgreSQL ke pengaturan konversi MySQL

PostgreSQL ke pengaturan konversi MySQL di DMS Schema Conversion meliputi:

- Komentar dalam kode SQL yang dikonversi: Pengaturan ini menyertakan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi. Pengaturan ini mendukung nilai-nilai berikut:
 - Kesalahan saja
 - Kesalahan dan peringatan
 - Semua pesan

Menyegarkan skema database Anda dalam Konversi Skema DMS

Setelah Anda membuat proyek migrasi, Konversi Skema DMS menyimpan informasi tentang skema sumber dan target Anda dalam proyek ini. Konversi Skema DMS menggunakan pemuatan lambat untuk memuat metadata hanya sesuai kebutuhan, seperti saat Anda memilih simpul di pohon database Anda. Anda dapat menggunakan eager loading untuk memuat informasi skema lebih cepat. Untuk melakukannya, pilih skema Anda, lalu pilih Muat metadata dari Tindakan.

Setelah Anda memuat objek secara otomatis atau manual ke proyek migrasi Anda, Konversi Skema DMS tidak menggunakan pemuatan lambat lagi. Jadi ketika Anda mengubah objek, seperti tabel dan prosedur dalam database Anda, pastikan untuk me-refresh mereka dalam proyek migrasi Anda.

Untuk menyegarkan skema dari database, pilih objek yang ingin disegarkan, dan pilih Refresh dari database dari Actions. Anda dapat menyegarkan objek database dalam skema basis data sumber dan target Anda:

- Sumber - Jika Anda memperbarui skema basis data sumber Anda, pilih Refresh dari database untuk mengganti skema dalam proyek Anda dengan skema terbaru dari database sumber Anda.
- Target - Jika Anda memperbarui skema untuk basis data target Anda, Konversi Skema DMS menggantikan skema dalam proyek Anda dengan skema terbaru dari basis data target Anda. Konversi Skema DMS menggantikan kode yang dikonversi dengan kode dari basis data target Anda. Pastikan Anda menerapkan kode yang dikonversi ke database target Anda sebelum memilih Refresh dari database. Jika tidak, ubah skema basis data sumber Anda lagi.

Menyimpan dan menerapkan kode konversi Anda dalam Konversi Skema DMS

Setelah Konversi Skema DMS mengonversi objek basis data sumber Anda, itu tidak segera menerapkan kode yang dikonversi ke database target Anda. Sebagai gantinya, DMS Schema Conversion menyimpan kode yang dikonversi dalam proyek Anda sampai Anda siap menerapkannya ke database target Anda.

Sebelum menerapkan kode yang dikonversi, Anda dapat memperbarui kode basis data sumber dan mengonversi objek yang diperbarui lagi untuk mengatasi item tindakan yang ada. Untuk informasi selengkapnya tentang item yang Konversi Skema DMS tidak dapat mengonversi secara otomatis, lihat [Membuat laporan penilaian migrasi database dengan Konversi Skema DMS](#) Untuk informasi selengkapnya tentang menyegarkan objek basis data sumber Anda dalam proyek migrasi untuk Konversi Skema DMS, lihat [Menyegarkan skema database Anda](#)

Alih-alih menerapkan kode yang dikonversi langsung ke database Anda di Konversi Skema DMS, Anda dapat menyimpan kode ke file sebagai skrip SQL. Anda dapat meninjau skrip SQL ini, mengeditnya jika perlu, dan kemudian secara manual menerapkan skrip SQL ini ke database target Anda.

Menyimpan kode yang telah dikonversi ke file SQL

Anda dapat menyimpan skema yang dikonversi sebagai skrip SQL dalam file teks. Anda dapat memodifikasi kode yang dikonversi untuk menangani item tindakan yang Konversi Skema DMS tidak dapat mengonversi secara otomatis. Anda kemudian dapat menjalankan skrip SQL yang diperbarui pada database target Anda untuk menerapkan kode yang dikonversi ke database target Anda.

Untuk menyimpan skema yang dikonversi sebagai skrip SQL

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.
3. Pilih proyek migrasi Anda, lalu pilih Konversi skema.
4. Pilih Luncurkan konversi skema. Halaman konversi Skema terbuka.
5. Di panel kanan, pilih skema basis data target atau pilih objek yang dikonversi yang ingin Anda simpan. Pastikan bahwa DMS Schema Conversion menyoroti nama node induk dengan warna biru dan mengaktifkan menu Actions untuk database target.
6. Pilih Simpan sebagai SQL untuk Tindakan. Kotak dialog Simpan muncul.
7. Pilih Simpan sebagai SQL untuk mengonfirmasi pilihan Anda.

Konversi Skema DMS membuat arsip dengan file SQL dan menyimpan arsip ini di bucket Amazon S3 Anda.

8. (Opsional) Ubah bucket S3 untuk arsip dengan mengedit pengaturan konversi skema di profil instans Anda.
9. Buka skrip SQL dari bucket S3 Anda.

Menerapkan kode konversi Anda

Ketika Anda siap untuk menerapkan kode yang dikonversi ke database target Anda, pilih objek database di panel kanan proyek Anda. Anda dapat menerapkan perubahan ke seluruh skema database atau objek skema database yang dipilih.

Setelah Anda memilih objek database, DMS Schema Conversion menyoroti nama node yang dipilih atau node induk berwarna biru. Kemudian mengaktifkan menu Tindakan. Pilih Terapkan perubahan untuk Tindakan. Di kotak dialog yang muncul, pilih Terapkan untuk mengonfirmasi pilihan Anda dan menerapkan kode yang dikonversi ke basis data target Anda.

Menerapkan skema paket ekstensi

Saat Anda menerapkan skema yang dikonversi ke database target untuk pertama kalinya, Konversi Skema DMS mungkin juga menerapkan skema paket ekstensi. Skema paket ekstensi mengemulasi fungsi sistem dari database sumber yang diperlukan untuk menjalankan kode yang dikonversi untuk basis data target Anda. Jika kode yang dikonversi menggunakan fungsi paket ekstensi, pastikan Anda menerapkan skema paket ekstensi.

Untuk menerapkan paket ekstensi ke database target Anda secara manual, pilih Terapkan perubahan untuk Tindakan. Di kotak dialog yang muncul, pilih konfirmasi untuk menerapkan paket ekstensi ke database target Anda.

Kami menyarankan Anda untuk tidak memodifikasi skema paket ekstensi untuk menghindari hasil yang tidak terduga dalam kode yang dikonversi.

Lihat informasi yang lebih lengkap di [Menggunakan paket ekstensi dalam Konversi Skema DMS](#).

Menggunakan paket ekstensi dalam Konversi Skema DMS

Paket ekstensi dalam Konversi Skema DMS adalah modul add-on yang mengemulasi fungsi database sumber yang tidak didukung dalam database target. Gunakan paket ekstensi untuk memastikan bahwa kode yang dikonversi menghasilkan hasil yang sama dengan kode sumber. Sebelum Anda dapat menginstal paket ekstensi, konversikan skema database Anda.

Setiap paket ekstensi menyertakan skema database. Skema ini mencakup fungsi SQL, prosedur, tabel, dan tampilan untuk meniru objek pemrosesan transaksi online tertentu (OLTP) atau fungsi bawaan yang tidak didukung dari database sumber.

Ketika Anda mengkonversi database sumber Anda, DMS Schema Conversion menambahkan skema tambahan ke database target Anda. Skema ini mengimplementasikan fungsi sistem SQL dari database sumber yang diperlukan untuk menjalankan kode Anda dikonversi pada database target Anda. Skema tambahan ini disebut skema paket ekstensi.

Skema paket ekstensi diberi nama sesuai dengan database sumber Anda sebagai berikut:

- Microsoft SQL Server – `aws_sqlserver_ext`
- Oracle – `aws_oracle_ext`

Anda dapat menerapkan paket ekstensi dengan dua cara:

- Konversi Skema DMS dapat secara otomatis menerapkan paket ekstensi saat Anda menerapkan kode yang dikonversi. DMS Schema Konversi berlaku paket ekstensi sebelum menerapkan semua objek skema lainnya.
- Anda dapat menerapkan paket ekstensi secara manual. Untuk melakukannya, pilih skema paket ekstensi di pohon database target Anda, lalu pilih Terapkan, lalu Terapkan paket ekstensi.

Migrasi database ke Amazon RDS yang setara dengan AWS DMS

Migrasi data homogen di AWS Database Migration Service (AWS DMS) menyederhanakan migrasi database lokal yang dikelola sendiri ke ekuivalen Amazon Relational Database Service (Amazon RDS). Misalnya, Anda dapat menggunakan migrasi data homogen untuk memigrasikan database PostgreSQL lokal ke Amazon RDS for PostgreSQL atau Aurora PostgreSQL. Untuk migrasi data yang homogen, AWS DMS gunakan alat database asli untuk menyediakan migrasi yang mudah dan berkinerja baik like-to-like .

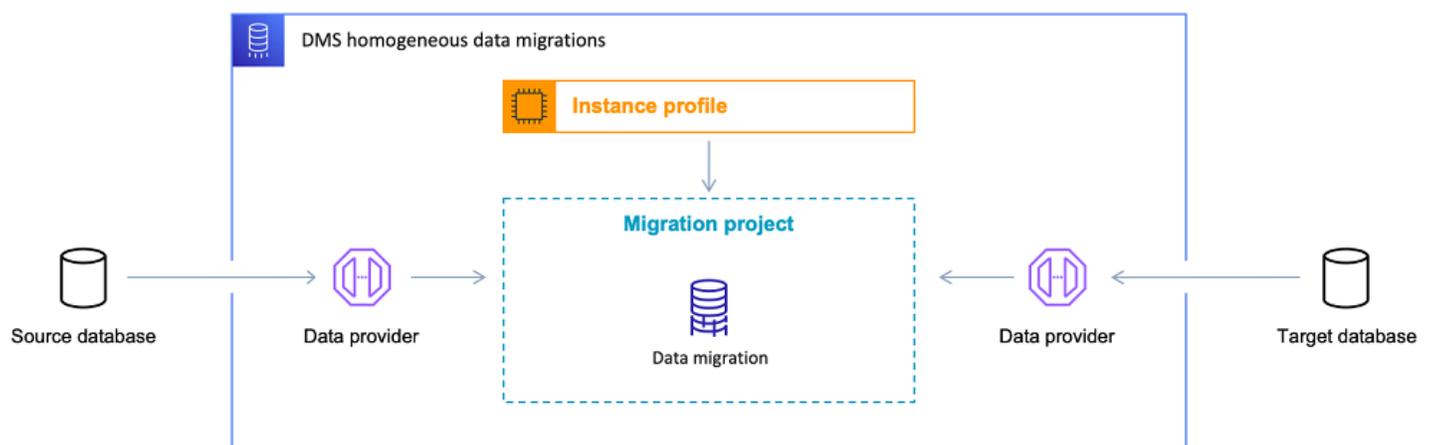
Migrasi data homogen adalah tanpa server, yang berarti bahwa AWS DMS secara otomatis menskalakan sumber daya yang diperlukan untuk migrasi Anda. Dengan migrasi data homogen, Anda dapat memigrasikan data, partisi tabel, tipe data, dan objek sekunder seperti fungsi, prosedur tersimpan, dan sebagainya.

Pada tingkat tinggi, migrasi data homogen beroperasi dengan profil instans, penyedia data, dan proyek migrasi. Saat Anda membuat proyek migrasi dengan sumber yang kompatibel dan penyedia data target dari jenis yang sama, AWS DMS menerapkan lingkungan tanpa server tempat migrasi data Anda berjalan. Selanjutnya, AWS DMS menghubungkan ke penyedia data sumber, membaca data sumber, membuang file pada disk, dan mengembalikan data menggunakan alat database asli. Untuk informasi selengkapnya tentang profil instans, penyedia data, dan proyek migrasi, lihat [Bekerja dengan penyedia data, profil instans, dan proyek migrasi di AWS DMS](#).

Untuk daftar database sumber yang didukung, lihat [Sumber untuk migrasi data homogen DMS](#).

Untuk daftar database target yang didukung, lihat [Target untuk migrasi data homogen DMS](#).

Diagram berikut menggambarkan bagaimana migrasi data homogen bekerja.



Bagian berikut memberikan informasi tentang penggunaan migrasi data homogen.

Topik

- [Wilayah AWS yang Didukung](#)
- [Fitur](#)
- [Keterbatasan untuk migrasi data homogen](#)
- [Ikhtisar proses migrasi data homogen di AWS DMS](#)
- [Menyiapkan migrasi data homogen di AWS DMS](#)
- [Membuat penyedia data sumber untuk migrasi data homogen di AWS DMS](#)
- [Membuat penyedia data target untuk migrasi data homogen di AWS DMS](#)
- [Menjalankan migrasi data homogen di AWS DMS](#)
- [Pemecahan masalah untuk migrasi data homogen di AWS DMS](#)

Wilayah AWS yang Didukung

Anda dapat menjalankan migrasi data homogen berikut Wilayah AWS ini.

Nama Wilayah	wilayah
AS Timur (Virginia Utara)	us-east-1
US East (Ohio)	us-east-2
US West (Oregon)	us-west-2
Asia Pasifik (Tokyo)	ap-northeast-1
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Europe (Frankfurt)	eu-central-1
Eropa (Stockholm)	eu-north-1
Eropa (Irlandia)	eu-west-1

Fitur

Migrasi data homogen menyediakan fitur-fitur berikut:

- AWS DMS secara otomatis mengelola sumber daya komputasi dan penyimpanan AWS Cloud yang diperlukan untuk migrasi data yang homogen. AWS DMS menyebarkan sumber daya ini di lingkungan tanpa server saat Anda memulai migrasi data.
- AWS DMS menggunakan alat database asli untuk memulai migrasi sepenuhnya otomatis antara database dari jenis yang sama.
- Anda dapat menggunakan migrasi data homogen untuk memigrasikan data Anda serta objek sekunder seperti partisi, fungsi, prosedur tersimpan, dan sebagainya.
- Anda dapat menjalankan migrasi data homogen dalam tiga mode migrasi berikut: beban penuh, replikasi berkelanjutan, dan beban penuh dengan replikasi berkelanjutan.
- Untuk migrasi data homogen, Anda dapat menggunakan database Amazon EC2, Amazon RDS lokal sebagai sumber. Anda dapat memilih Amazon RDS atau Amazon Aurora sebagai target migrasi untuk migrasi data homogen.

Keterbatasan untuk migrasi data homogen

Batasan berikut berlaku saat Anda menggunakan migrasi data homogen:

- Migrasi data homogen tidak mendukung aturan pemilihan. Selain itu, Anda tidak dapat menggunakan aturan transformasi untuk mengubah tipe data kolom, memindahkan objek dari satu skema ke skema lainnya, atau mengubah nama objek.
- Migrasi data homogen tidak menyediakan alat bawaan untuk validasi data.
- Saat menggunakan migrasi data homogen dengan PostgreSQL, memigrasikan tampilan sebagai tabel ke AWS DMS database target Anda.
- Migrasi data homogen tidak menangkap perubahan tingkat skema selama replikasi data yang sedang berlangsung. Jika Anda membuat tabel baru di database sumber Anda, maka tidak AWS DMS dapat memigrasikan tabel ini. Untuk memigrasikan tabel baru ini, mulai ulang migrasi data Anda.
- Anda tidak dapat menggunakan migrasi data homogen AWS DMS untuk memigrasikan data dari versi database yang lebih tinggi ke versi database yang lebih rendah.
- Anda tidak dapat menggunakan migrasi data homogen di CLI atau API.

- Migrasi data homogen tidak mendukung pembuatan koneksi dengan instance database dalam rentang CIDR sekunder VPC.
- Anda tidak dapat menggunakan port 8081 untuk migrasi homogen dari penyedia data Anda.

Ikhtisar proses migrasi data homogen di AWS DMS

Anda dapat menggunakan migrasi data homogen AWS DMS untuk memigrasi data antara dua database dengan jenis yang sama. Gunakan alur kerja berikut untuk membuat dan menjalankan migrasi data.

1. Buat kebijakan dan peran yang diperlukan AWS Identity and Access Management (IAM). Untuk informasi selengkapnya, lihat [Membuat sumber daya IAM](#).
2. Konfigurasi database sumber dan target Anda dan buat pengguna database dengan izin minimum yang diperlukan untuk migrasi data homogen di AWS DMS. Untuk informasi selengkapnya, lihat [Membuat penyedia data sumber](#) dan [Membuat penyedia data target](#).
3. Simpan sumber dan target kredensi database Anda di AWS Secrets Manager. Untuk informasi selengkapnya, lihat [Langkah 1: Buat rahasia](#) di Panduan AWS Secrets Manager Pengguna.
4. Buat grup subnet, profil instans, dan penyedia data di AWS DMS konsol. Untuk informasi lebih lanjut, lihat [Membuat grup subnet](#), [Membuat profil instance](#), dan [Membuat penyedia data](#).
5. Buat proyek migrasi dengan menggunakan sumber daya yang Anda buat di langkah sebelumnya. Untuk informasi selengkapnya, lihat [Membuat proyek migrasi](#).
6. Buat, konfigurasi, dan mulai migrasi data. Untuk informasi selengkapnya, lihat [Membuat migrasi data](#).
7. Setelah Anda menyelesaikan beban penuh atau replikasi yang sedang berlangsung, Anda dapat memotong untuk mulai menggunakan database target baru Anda.
8. Bersihkan sumber daya Anda. Amazon mengakhiri migrasi data Anda dalam proyek migrasi Anda dalam tiga hari setelah Anda menyelesaikan migrasi. Namun, Anda perlu menghapus sumber daya seperti profil instans, penyedia data, kebijakan dan peran IAM secara manual, dan rahasia di dalamnya AWS Secrets Manager.

Untuk informasi selengkapnya tentang migrasi data homogen AWS DMS, baca panduan step-by-step migrasi untuk [PostgreSQL ke Amazon RDS untuk migrasi PostgreSQL](#).

[Video ini](#) memperkenalkan migrasi data homogen AWS DMS dan membantu Anda membiasakan diri dengan fitur ini.

Menyiapkan migrasi data homogen di AWS DMS

Untuk mengatur migrasi data homogen AWS DMS, selesaikan tugas prasyarat berikut.

Topik

- [Membuat sumber daya IAM yang diperlukan untuk migrasi data homogen di AWS DMS](#)
- [Menyiapkan jaringan untuk migrasi data homogen di AWS DMS](#)

Membuat sumber daya IAM yang diperlukan untuk migrasi data homogen di AWS DMS

Untuk menjalankan migrasi data homogen, Anda harus membuat kebijakan IAM dan peran IAM di akun Anda untuk berinteraksi dengan layanan lain. AWS Di bagian ini, Anda membuat sumber daya IAM yang diperlukan ini.

Topik

- [Membuat kebijakan IAM untuk migrasi data homogen di AWS DMS](#)
- [Membuat peran IAM untuk migrasi data homogen di AWS DMS](#)

Membuat kebijakan IAM untuk migrasi data homogen di AWS DMS

Untuk mengakses database Anda dan memigrasi data, AWS DMS buat lingkungan tanpa server untuk migrasi data homogen. Dalam lingkungan ini, AWS DMS memerlukan akses ke VPC peering, tabel rute, kelompok keamanan, dan sumber daya lainnya AWS. Selain itu, AWS DMS menyimpan log, metrik, dan kemajuan untuk setiap migrasi data di AmazonCloudWatch. Untuk membuat proyek migrasi data, AWS DMS perlu akses ke layanan ini.

Pada langkah ini, Anda membuat kebijakan IAM yang menyediakan AWS DMS akses ke Amazon EC2 dan CloudWatch sumber daya. Selanjutnya, buat peran IAM dan lampirkan kebijakan ini.

Untuk membuat kebijakan IAM untuk migrasi data homogen di AWS DMS

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat kebijakan.

4. Di halaman Buat kebijakan, pilih tab JSON.
5. Tempel JSON berikut ke editor.

```
{
  "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. Pilih Berikutnya: Tag dan Berikutnya: Ulasan.
7. Masukkan **HomogeneousDataMigrationsPolicy** untuk Nama*, dan pilih Buat kebijakan.

Membuat peran IAM untuk migrasi data homogen di AWS DMS

Pada langkah ini, Anda membuat peran IAM yang menyediakan AWS DMS akses ke AWS Secrets Manager, Amazon EC2, dan CloudWatch

Untuk membuat peran IAM untuk migrasi data homogen di AWS DMS

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Peran.
3. Pilih Buat peran.
4. Pada halaman Pilih entitas tepercaya, untuk jenis entitas tepercaya, pilih AWS Layanan. Untuk Gunakan kasus untuk AWS layanan lain, pilih DMS.
5. Pilih kotak centang DMS dan pilih Berikutnya.
6. Pada halaman Tambahkan izin, pilih HomogeneousDataMigrationsPolicy yang Anda buat sebelumnya. Juga, pilih SecretsManagerReadWrite. Pilih Selanjutnya.
7. Pada halaman Nama, tinjau, dan buat, masukkan **HomogeneousDataMigrationsRole** nama Peran, lalu pilih Buat peran.
8. Pada halaman Peran, masukkan nama **HomogeneousDataMigrationsRole** Peran. Pilih HomogeneousDataMigrationsRole.

9. Pada `HomogeneousDataMigrationsRole` halaman, pilih tab Hubungan kepercayaan. Pilih Edit kebijakan kepercayaan.
10. Pada halaman Edit kebijakan kepercayaan, rekatkan JSON berikut ke editor, ganti teks yang ada.

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

Dalam contoh sebelumnya, ganti *your_region* dengan nama Anda. Wilayah AWS

Kebijakan berbasis sumber daya sebelumnya memberikan izin AWS DMS kepada prinsipal layanan untuk melakukan tugas sesuai dengan kebijakan yang dikelola dan dikelola pelanggan. `AWS SecretsManagerReadWriteHomogeneousDataMigrationsPolicy`

11. Pilih Buat Kebijakan.

Menyiapkan jaringan untuk migrasi data homogen di AWS DMS

AWS DMS menciptakan lingkungan tanpa server untuk migrasi data homogen di cloud pribadi virtual (VPC) berdasarkan layanan Amazon VPC. Saat membuat profil instans, Anda menentukan VPC yang akan digunakan. Anda dapat menggunakan VPC default untuk akun Anda dan Wilayah AWS, atau Anda dapat membuat VPC baru.

Untuk setiap migrasi data, AWS DMS buat koneksi peering VPC dengan VPC yang Anda gunakan untuk profil instans Anda. Selanjutnya, AWS DMS tambahkan blok CIDR di grup keamanan yang terkait dengan profil instans Anda. Karena AWS DMS melampirkan alamat IP publik ke profil instans

Anda, semua migrasi data yang menggunakan profil instans yang sama memiliki alamat IP publik yang sama. Saat migrasi data Anda berhenti atau gagal, AWS DMS hapus koneksi peering VPC.

Untuk menghindari blok CIDR tumpang tindih dengan VPC profil instans VPC Anda, AWS DMS gunakan /24 awalan dari salah satu blok CIDR berikut:., dan. 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 Misalnya, jika Anda menjalankan tiga migrasi data secara paralel, AWS DMS gunakan blok CIDR berikut untuk membuat koneksi peering VPC.

- 192.168.0.0/24— untuk migrasi data pertama
- 192.168.1.0/24— untuk migrasi data kedua
- 192.168.2.0/24— untuk migrasi data ketiga

Anda dapat menggunakan konfigurasi jaringan yang berbeda untuk mengatur interaksi antara sumber dan basis data target Anda. AWS DMS Selain itu, untuk replikasi data yang sedang berlangsung, Anda harus mengatur interaksi antara database sumber dan target Anda. Konfigurasi ini bergantung pada lokasi penyedia data sumber dan pengaturan jaringan Anda. Bagian berikut memberikan deskripsi konfigurasi jaringan umum.

Topik

- [Menggunakan VPC tunggal untuk penyedia sumber dan target data](#)
- [Menggunakan VPC yang berbeda untuk penyedia sumber dan target data](#)
- [Menggunakan penyedia data sumber lokal](#)
- [Mengkonfigurasi replikasi data yang sedang berlangsung](#)

Menggunakan VPC tunggal untuk penyedia sumber dan target data

Dalam konfigurasi ini, AWS DMS sambungkan ke penyedia data sumber dan target Anda dalam jaringan pribadi.

Untuk mengonfigurasi jaringan saat penyedia data sumber dan target Anda berada di VPC yang sama

1. Buat grup subnet di AWS DMS konsol dengan VPC dan subnet yang digunakan penyedia data sumber dan target Anda. Untuk informasi selengkapnya, lihat [Membuat grup subnet](#).

2. Buat profil instans di AWS DMS konsol dengan VPC dan grup subnet yang Anda buat. Selain itu, pilih grup keamanan VPC yang digunakan penyedia data sumber dan target Anda. Untuk informasi selengkapnya, lihat [Membuat profil instance](#).

Konfigurasi ini tidak mengharuskan Anda untuk menggunakan alamat IP publik untuk migrasi data.

Menggunakan VPC yang berbeda untuk penyedia sumber dan target data

Dalam konfigurasi ini, AWS DMS gunakan jaringan pribadi untuk terhubung ke sumber atau penyedia data target Anda. Untuk penyedia data lain, AWS DMS gunakan jaringan publik. Bergantung pada penyedia data yang Anda miliki di VPC yang sama dengan profil instans Anda, pilih salah satu dari konfigurasi berikut.

Untuk mengkonfigurasi jaringan pribadi untuk penyedia data sumber Anda dan jaringan publik untuk penyedia data target Anda

1. Buat grup subnet di AWS DMS konsol dengan VPC dan subnet yang digunakan penyedia data sumber Anda. Untuk informasi selengkapnya, lihat [Membuat grup subnet](#).
2. Buat profil instans di AWS DMS konsol dengan VPC dan grup subnet yang Anda buat. Selain itu, pilih grup keamanan VPC yang digunakan penyedia data sumber Anda. Untuk informasi selengkapnya, lihat [Membuat profil instance](#).
3. Buka proyek migrasi Anda. Pada tab Migrasi data, pilih migrasi data Anda. Catat alamat IP publik di bawah Konektivitas dan keamanan pada tab Detail.
4. Izinkan akses dari alamat IP publik migrasi data Anda di grup keamanan database target Anda. Untuk informasi selengkapnya, lihat [Mengontrol akses dengan grup keamanan](#) di Panduan Pengguna Amazon Relational Database Service.

Untuk mengonfigurasi jaringan publik untuk penyedia data sumber Anda dan jaringan pribadi untuk penyedia data target Anda

1. Buat grup subnet di AWS DMS konsol dengan VPC dan subnet yang digunakan penyedia data target Anda. Untuk informasi selengkapnya, lihat [Membuat grup subnet](#).
2. Buat profil instans di AWS DMS konsol dengan VPC dan grup subnet yang Anda buat. Selain itu, pilih grup keamanan VPC yang digunakan penyedia data target Anda. Untuk informasi selengkapnya, lihat [Membuat profil instance](#).
3. Buka proyek migrasi Anda. Pada tab Migrasi data, pilih migrasi data Anda. Catat alamat IP publik di bawah Konektivitas dan keamanan pada tab Detail.

4. Izinkan akses dari alamat IP publik migrasi data Anda di grup keamanan database sumber Anda. Untuk informasi selengkapnya, lihat [Mengontrol akses dengan grup keamanan](#) di Panduan Pengguna Amazon Relational Database Service.

Menggunakan penyedia data sumber lokal

Dalam konfigurasi ini, AWS DMS sambungkan ke penyedia data sumber Anda dalam jaringan publik. AWS DMS menggunakan jaringan pribadi untuk terhubung ke penyedia data target Anda.

Mengonfigurasi jaringan untuk penyedia data lokal sumber

1. Buat grup subnet di AWS DMS konsol dengan VPC dan subnet yang digunakan penyedia data target Anda. Untuk informasi selengkapnya, lihat [Membuat grup subnet](#).
2. Buat profil instans di AWS DMS konsol dengan VPC dan grup subnet yang Anda buat. Selain itu, pilih grup keamanan VPC yang digunakan penyedia data target Anda. Untuk informasi selengkapnya, lihat [Membuat profil instance](#).
3. Buka proyek migrasi Anda. Pada tab Migrasi data, pilih migrasi data Anda. Catat alamat IP publik di bawah Konektivitas dan keamanan pada tab Detail.
4. Izinkan akses ke database sumber Anda dari alamat IP publik migrasi data Anda di AWS DMS.

AWS DMS membuat aturan masuk atau keluar di dalam grup keamanan VPC. Pastikan Anda tidak menghapus aturan ini karena tindakan ini dapat menyebabkan kegagalan migrasi data Anda. Anda dapat mengonfigurasi aturan Anda sendiri di grup keamanan VPC. Kami menyarankan Anda untuk menambahkan deskripsi ke aturan Anda sehingga Anda dapat mengelolanya.

Mengkonfigurasi replikasi data yang sedang berlangsung

Untuk menjalankan migrasi data dari tipe Full load and change data capture (CDC) atau Change data capture (CDC), Anda harus mengizinkan koneksi antara database sumber dan target Anda.

Untuk mengonfigurasi koneksi antara sumber dan basis data target yang dapat diakses publik

1. Catat alamat IP publik dari sumber dan basis data target Anda.
2. Izinkan akses ke database sumber Anda dari alamat IP publik database target Anda.
3. Izinkan akses ke database target Anda dari alamat IP publik database sumber Anda.

Untuk mengonfigurasi koneksi antara database sumber dan target Anda yang dapat diakses secara pribadi dalam satu VPC

1. Catat alamat IP pribadi sumber dan basis data target Anda.

⚠ Important

Jika database sumber dan target Anda berada di VPC yang berbeda atau di jaringan yang berbeda, maka Anda hanya dapat menggunakan alamat IP publik untuk sumber dan basis data target Anda. Anda hanya dapat menggunakan nama host publik atau alamat IP di penyedia data.

2. Izinkan akses ke database sumber Anda dari alamat IP pribadi database target Anda.
3. Izinkan akses ke database target Anda dari alamat IP pribadi database sumber Anda.

Membuat penyedia data sumber untuk migrasi data homogen di AWS DMS

Anda dapat menggunakan database yang kompatibel dengan MySQL, PostgreSQL, dan MongoDB yang kompatibel sebagai penyedia data sumber untuk in. [Migrasi data homogen AWS DMS](#)

Untuk versi database yang didukung, lihat [Penyedia data sumber untuk migrasi data homogen DMS](#).

Penyedia data sumber Anda dapat berupa database lokal, Amazon EC2, atau Amazon RDS.

Topik

- [Menggunakan database MySQL kompatibel sebagai sumber untuk migrasi data homogen di AWS DMS](#)
- [Menggunakan database PostgreSQL sebagai sumber migrasi data homogen AWS DMS](#)
- [Menggunakan database kompatibel MongoDB sebagai sumber untuk migrasi data homogen di AWS DMS](#)

Menggunakan database MySQL kompatibel sebagai sumber untuk migrasi data homogen di AWS DMS

Anda dapat menggunakan database MySQL-kompatibel (MySQL atau MariaDB) sebagai sumber untuk di [Migrasi data homogen](#) AWS DMS. Dalam hal ini, penyedia data sumber Anda dapat berupa database lokal, Amazon EC2, atau RDS untuk MySQL atau MariaDB.

Untuk menjalankan migrasi data homogen, Anda harus menggunakan pengguna database dengan SELECT hak istimewa untuk semua tabel sumber dan objek sekunder untuk replikasi. Untuk tugas change data capture (CDC), pengguna ini juga harus memiliki REPLICATION CLIENT (BINLOG MONITOR untuk versi MariaDB lebih lambat dari 10.5.2) dan hak istimewa. REPLICATION SLAVE. Untuk migrasi data muatan penuh, Anda tidak memerlukan dua hak istimewa ini.

Gunakan skrip berikut untuk membuat pengguna database dengan izin yang diperlukan dalam database MySQL Anda. Jalankan GRANT kueri untuk semua database tempat Anda bermigrasi. 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'@'%;
```

Pada contoh sebelumnya, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri. Jika versi database MySQL sumber Anda lebih rendah dari 8.0, maka Anda dapat melewati perintah. GRANT BACKUP_ADMIN

Gunakan skrip berikut untuk membuat pengguna database dengan izin yang diperlukan dalam database MariaDB Anda. Jalankan kueri GRANT untuk semua database tempat Anda bermigrasi. AWS

```
CREATE USER 'your_user'@'%' IDENTIFIED BY 'your_password';
GRANT SELECT, RELOAD, LOCK TABLES, REPLICATION SLAVE, BINLOG MONITOR, SHOW VIEW ON *.*
TO 'your_user'@'%;
```

Pada contoh sebelumnya, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri.

Bagian berikut menjelaskan prasyarat konfigurasi khusus untuk database MySQL yang dikelola sendiri dan AWS dikelola sendiri.

Topik

- [Menggunakan database MySQL yang kompatibel dengan dikelola sendiri sebagai sumber migrasi data yang homogen](#)
- [Menggunakan database kompatibel MySQL AWS -managed sebagai sumber untuk migrasi data homogen di AWS DMS](#)
- [Keterbatasan untuk menggunakan database MySQL yang kompatibel sebagai sumber migrasi data homogen](#)

Menggunakan database MySQL yang kompatibel dengan dikelola sendiri sebagai sumber migrasi data yang homogen

Bagian ini menjelaskan cara mengonfigurasi database kompatibel MySQL yang dihosting di tempat atau di instans Amazon EC2.

Periksa versi database MySQL atau MariaDB sumber Anda. Pastikan itu AWS DMS mendukung versi database MySQL atau MariaDB sumber Anda seperti yang dijelaskan dalam. [Sumber untuk migrasi data homogen DMS](#)

Untuk menggunakan CDC, pastikan untuk mengaktifkan binary logging. Untuk mengaktifkan logging biner, konfigurasi parameter berikut dalam file `my.ini` (Windows) atau `my.cnf` (UNIX) dari database MySQL atau MariaDB Anda.

Parameter	Nilai
<code>server-id</code>	Atur parameter ini supaya nilainya 1 atau lebih besar.
<code>log-bin</code>	Atur jalur ke berkas log biner, seperti <code>log-bin=E:\MySql_Logs\BinLog</code> . Jangan sertakan ekstensi file.
<code>binlog_format</code>	Atur parameter ini menjadi ROW. Kami merekomendasikan pengaturan ini selama replikasi karena dalam kasus-kasus tertentu ketika <code>binlog_format</code> diatur menjadi STATEMENT , inkonsistensi dapat terjadi ketika mereplikasi data ke target. Mesin database juga menulis data yang tidak konsisten serupa dengan target ketika <code>binlog_format</code> diatur ke MIXED, karena mesin database secara otomatis beralih ke logging STATEMENT berbasis.

Parameter	Nilai
<code>expire_logs_days</code>	Atur parameter ini supaya nilainya 1 atau lebih besar. Untuk mencegah penggunaan ruang disk secara berlebihan, kami sarankan Anda tidak menggunakan nilai default 0.
<code>binlog_checksum</code>	Atur parameter ini menjadi NONE.
<code>binlog_row_image</code>	Atur parameter ini menjadi FULL.
<code>log_slave_updates</code>	Tetapkan parameter ini TRUE jika Anda menggunakan replika MySQL atau MariaDB sebagai sumber.

Menggunakan database kompatibel MySQL AWS -managed sebagai sumber untuk migrasi data homogen di AWS DMS

Bagian ini menjelaskan cara mengonfigurasi Amazon RDS untuk MySQL dan Amazon RDS untuk instance database Amazon RDS for MariaDB.

Bila Anda menggunakan database MySQL atau MariaDB AWS -managed sebagai sumber AWS DMS migrasi data homogen, pastikan Anda memiliki prasyarat berikut untuk CDC:

- Untuk mengaktifkan log biner untuk RDS untuk MySQL dan MariaDB, aktifkan backup otomatis pada tingkat instans. Untuk mengaktifkan log biner untuk cluster MySQL Aurora, ubah variabel dalam grup parameter. `binlog_format` Anda tidak perlu mengaktifkan backup otomatis untuk cluster Aurora MySQL.

Selanjutnya, atur `binlog_format` parameter keROW.

Untuk informasi selengkapnya tentang menyiapkan pencadangan otomatis, lihat [Mengaktifkan pencadangan otomatis](#) di Panduan Pengguna Amazon RDS.

Untuk informasi selengkapnya tentang menyiapkan pencatatan biner untuk database Amazon RDS for MySQL atau MariaDB, [lihat Menyetel format pencatatan biner di](#) Panduan Pengguna Amazon RDS.

Untuk informasi selengkapnya tentang menyiapkan pencatatan biner untuk klaster MySQL Aurora, lihat [Bagaimana cara mengaktifkan pencatatan biner untuk klaster MySQL Amazon Aurora saya?](#) .

- Pastikan bahwa log biner tersedia untuk AWS DMS. Karena database MySQL dan MariaDB AWS -managed membersihkan log biner sesegera mungkin, Anda harus menambah lamanya waktu log tetap tersedia. Misalnya, untuk meningkatkan retensi log hingga 24 jam, jalankan perintah berikut.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

- Atur parameter `binlog_row_image` menjadi `Full`.
- Tetapkan parameter `binlog_checksum` ke `NONE`.
- Jika Anda menggunakan replika Amazon RDS MySQL atau MariaDB sebagai sumber, aktifkan pencadangan pada replika baca, dan pastikan parameter disetel ke `log_slave_updates TRUE`

Keterbatasan untuk menggunakan database MySQL yang kompatibel sebagai sumber migrasi data homogen

Batasan berikut berlaku saat menggunakan database MySQL yang kompatibel sebagai sumber migrasi data homogen:

- Objek MariaDB seperti urutan tidak didukung dalam tugas migrasi homogen.
- Migrasi dari MariaDB ke Amazon RDS MySQL/Aurora MySQL mungkin gagal karena perbedaan objek yang tidak kompatibel.
- Nama pengguna yang Anda gunakan untuk terhubung ke sumber data Anda memiliki batasan berikut:
 - Panjangnya bisa 2 hingga 64 karakter.
 - Tidak dapat memiliki spasi.
 - Dapat menyertakan karakter berikut: a-z, A-Z, 0-9, garis bawah (_).
 - Harus dimulai dengan a-z atau A-Z.
- Kata sandi yang Anda gunakan untuk menyambung ke sumber data Anda memiliki batasan berikut:
 - Panjangnya bisa 1 hingga 128 karakter.
 - Tidak dapat memuat salah satu dari berikut ini: kutipan tunggal ('), kutipan ganda ("), titik koma (;) atau spasi.

Menggunakan database PostgreSQL sebagai sumber migrasi data homogen AWS DMS

Anda dapat menggunakan database PostgreSQL sebagai sumber untuk in. [Migrasi data homogen AWS DMS](#) Dalam hal ini, penyedia data sumber Anda dapat berupa database lokal, Amazon EC2, atau RDS untuk PostgreSQL.

Untuk menjalankan migrasi data homogen, berikan izin superuser untuk pengguna database yang Anda tentukan untuk database sumber PostgreSQL AWS DMS Anda. Pengguna database memerlukan izin superuser untuk mengakses fungsi spesifik replikasi di sumbernya. Untuk migrasi data muatan penuh, pengguna database Anda memerlukan SELECT izin pada tabel untuk memigrasinya.

Gunakan skrip berikut untuk membuat pengguna database dengan izin yang diperlukan dalam database sumber PostgreSQL Anda. Jalankan GRANT kueri untuk semua database tempat Anda bermigrasi. 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;
```

Pada contoh sebelumnya, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri.

Bagian berikut menjelaskan prasyarat konfigurasi spesifik untuk database PostgreSQL yang dikelola sendiri dan AWS dikelola sendiri.

Topik

- [Menggunakan database PostgreSQL yang dikelola sendiri sebagai sumber migrasi data homogen di AWS DMS](#)
- [Menggunakan database PostgreSQL AWS -managed sebagai sumber migrasi data homogen di AWS DMS](#)
- [Keterbatasan untuk menggunakan database yang kompatibel dengan PostgreSQL sebagai sumber migrasi data homogen](#)

Menggunakan database PostgreSQL yang dikelola sendiri sebagai sumber migrasi data homogen di AWS DMS

Bagian ini menjelaskan cara mengonfigurasi database PostgreSQL yang dihosting lokal atau di instans Amazon EC2.

Periksa versi database PostgreSQL sumber Anda. Pastikan itu AWS DMS mendukung versi database PostgreSQL sumber Anda seperti yang dijelaskan dalam [Sumber untuk migrasi data homogen DMS](#)

Migrasi data homogen mendukung pengambilan data perubahan (CDC) menggunakan replikasi logis. Untuk mengaktifkan replikasi logis pada database sumber PostgreSQL yang dikelola sendiri, tetapkan parameter dan nilai berikut dalam file konfigurasi: `postgresql.conf`

- Atur `wal_level` ke `logical`.
- Tetapkan `max_replication_slots` untuk nilai yang lebih besar dari 1.

Tetapkan nilai `max_replication_slots` sesuai dengan jumlah tugas yang ingin Anda jalankan. Misalnya, untuk menjalankan lima tugas Anda menetapkan minimal lima slot. Slot terbuka secara otomatis segera setelah tugas dimulai dan tetap terbuka bahkan ketika tugas tidak lagi berjalan. Pastikan Anda menghapus slot terbuka secara manual.

- Atur `max_wal_senders` menjadi nilai yang lebih besar dari 1.

Parameter `max_wal_senders` mengatur jumlah tugas bersamaan yang dapat berjalan.

- Parameter `wal_sender_timeout` mengakhiri sambungan replikasi yang tidak aktif lebih lama dari jumlah milidetik tertentu. Standarnya adalah 60000 milidetik (60 detik). Menyetel nilai ke 0 (no) menonaktifkan mekanisme batas waktu, dan merupakan pengaturan yang valid untuk DMS.

Beberapa parameter bersifat statis, dan Anda hanya dapat mengaturnya pada awal server. Setiap perubahan pada entri mereka dalam file konfigurasi diabaikan sampai server dimulai ulang. Untuk informasi selengkapnya, lihat [dokumentasi PostgreSQL](#).

Menggunakan database PostgreSQL AWS -managed sebagai sumber migrasi data homogen di AWS DMS

Bagian ini menjelaskan cara mengonfigurasi Amazon RDS untuk instance database Amazon RDS for PostgreSQL.

Gunakan akun pengguna AWS master untuk instance PostgreSQL DB sebagai akun pengguna untuk penyedia data sumber PostgreSQL untuk migrasi data homogen di. AWS DMS Akun pengguna utama memiliki peran yang diperlukan yang memungkinkan untuk mengatur CDC. Jika Anda menggunakan akun selain akun pengguna utama, maka akun tersebut harus memiliki `rds_superuser` peran dan `rds_replication` peran tersebut. Peran `rds_replication` memberikan izin untuk mengelola slot logis dan mengalirkan data menggunakan slot logis.

Gunakan contoh kode berikut memberikan `rds_replication` peran `rds_superuser` dan.

```
GRANT rds_superuser to your_user;  
GRANT rds_replication to your_user;
```

Pada contoh sebelumnya, ganti *your_user* dengan *nama pengguna* database Anda.

Untuk mengaktifkan replikasi logis, atur `rds.logical_replication` parameter dalam grup parameter DB Anda ke 1. Parameter statis memerlukan reboot dari instans DB agar menjadi berpengaruh.

Keterbatasan untuk menggunakan database yang kompatibel dengan PostgreSQL sebagai sumber migrasi data homogen

Batasan berikut berlaku saat menggunakan database yang kompatibel dengan PostgreSQL sebagai sumber migrasi data homogen:

- Nama pengguna yang Anda gunakan untuk terhubung ke sumber data Anda memiliki batasan berikut:
 - Panjangnya bisa 2 hingga 64 karakter.
 - Tidak dapat memiliki spasi.
 - Dapat menyertakan karakter berikut: a-z, A-Z, 0-9, garis bawah (_).
 - Harus dimulai dengan a-z atau A-Z.
- Kata sandi yang Anda gunakan untuk menyambung ke sumber data Anda memiliki batasan berikut:
 - Panjangnya bisa 1 hingga 128 karakter.
 - Tidak dapat memuat salah satu dari berikut ini: kutipan tunggal ('), kutipan ganda ("), titik koma (;) atau spasi.

Menggunakan database kompatibel MongoDB sebagai sumber untuk migrasi data homogen di AWS DMS

Anda dapat menggunakan database yang kompatibel dengan MongoDB sebagai sumber migrasi data Homogen di AWS DMS. Dalam hal ini, penyedia data sumber Anda dapat berupa database Amazon EC2 untuk MongoDB lokal, atau Amazon DocumentDB (dengan kompatibilitas MongoDB).

Untuk versi database yang didukung, lihat [Penyedia data sumber untuk migrasi data homogen DMS](#).

Bagian berikut menjelaskan prasyarat konfigurasi khusus untuk database MongoDB yang dikelola sendiri dan database Amazon DocumentDB yang dikelola sendiri. AWS

Topik

- [Menggunakan database MongoDB yang dikelola sendiri sebagai sumber migrasi data homogen di AWS DMS](#)
- [Menggunakan database Amazon DocumentDB sebagai sumber migrasi data homogen di AWS DMS](#)
- [Fitur untuk menggunakan database yang kompatibel dengan MongoDB sebagai sumber migrasi data homogen](#)
- [Keterbatasan untuk menggunakan database yang kompatibel dengan MongoDB sebagai sumber migrasi data homogen](#)
- [Praktik terbaik untuk menggunakan database yang kompatibel dengan MongoDB sebagai sumber migrasi data homogen](#)

Menggunakan database MongoDB yang dikelola sendiri sebagai sumber migrasi data homogen di AWS DMS

Bagian ini menjelaskan cara mengonfigurasi database MongoDB yang dihosting lokal atau di instans Amazon EC2.

Periksa versi database MongoDB sumber Anda. Pastikan itu AWS DMS mendukung versi database MongoDB sumber Anda seperti yang dijelaskan dalam [Penyedia data sumber untuk migrasi data homogen DMS](#)

Untuk menjalankan migrasi data homogen dengan sumber MongoDB, Anda dapat membuat akun pengguna dengan hak akses root, atau pengguna dengan izin hanya pada database untuk

bermigrasi. Untuk informasi selengkapnya tentang pembuatan pengguna, lihat [Izin yang dibutuhkan saat menggunakan MongoDB sebagai sumber untuk AWS DMS](#).

Untuk menggunakan replikasi berkelanjutan atau CDC dengan MongoDB, memerlukan AWS DMS akses ke log operasi MongoDB (oplog). Untuk informasi selengkapnya, lihat [Mengonfigurasi rangkaian replika MongoDB yang untuk CDC](#).

Untuk informasi tentang metode otentikasi MongoDB, lihat [Persyaratan keamanan saat menggunakan MongoDB sebagai sumber untuk AWS DMS](#)

Untuk MongoDB sebagai sumber, migrasi data homogen mendukung semua tipe data yang didukung Amazon DocumentDB.

Untuk MongoDB sebagai sumber, untuk menyimpan kredensial pengguna di Secrets Manager, Anda perlu menyediakannya dalam teks biasa, menggunakan jenis rahasia lainnya. Untuk informasi selengkapnya, lihat [Menggunakan rahasia untuk mengakses titik akhir AWS Database Migration Service](#).

Contoh kode berikut menunjukkan bagaimana untuk menyimpan rahasia database menggunakan teks biasa.

```
{
  "username": "dbuser",
  "password": "dbpassword"
}
```

Menggunakan database Amazon DocumentDB sebagai sumber migrasi data homogen di AWS DMS

Bagian ini menjelaskan cara mengonfigurasi instans database Amazon DocumentDB Anda untuk digunakan sebagai sumber migrasi data yang homogen.

Gunakan nama pengguna utama untuk instans Amazon DocumentDB sebagai akun pengguna untuk penyedia data sumber yang kompatibel dengan MongoDB untuk migrasi data homogen. AWS DMS Akun pengguna utama memiliki peran yang diperlukan yang memungkinkan untuk mengatur CDC. Jika Anda menggunakan akun selain akun pengguna utama, maka akun tersebut harus memiliki peran root. Untuk informasi selengkapnya tentang pembuatan pengguna sebagai akun root, lihat [Menetapkan izin untuk menggunakan Amazon DocumentDB sebagai sumber](#).

Untuk mengaktifkan replikasi logis, atur `change_stream_log_retention_duration` parameter dalam grup parameter database Anda ke pengaturan yang sesuai untuk beban kerja transaksi Anda.

Mengubah parameter statis ini mengharuskan Anda untuk me-reboot instans DB Anda agar berlaku. Sebelum memulai migrasi data untuk semua jenis tugas termasuk Hanya Beban Penuh, aktifkan aliran perubahan Amazon DocumentDB untuk semua koleksi dalam database tertentu, atau hanya untuk koleksi yang dipilih. Untuk informasi selengkapnya tentang mengaktifkan aliran perubahan untuk Amazon DocumentDB, [lihat Mengaktifkan Aliran Perubahan di](#) panduan pengembang Amazon DocumentDB.

Note

AWS DMS menggunakan aliran perubahan Amazon DocumentDB untuk menangkap perubahan selama replikasi yang sedang berlangsung. Jika Amazon DocumentDB menghapus catatan dari aliran perubahan sebelum DMS membacanya, tugas Anda akan gagal. Kami merekomendasikan pengaturan `change_stream_log_retention_duration` parameter untuk mempertahankan perubahan setidaknya selama 24 jam.

Untuk menggunakan Amazon DocumentDB untuk migrasi data homogen, simpan kredensi pengguna di Secrets Manager di bawah database Credentials for Amazon DocumentDB.

Fitur untuk menggunakan database yang kompatibel dengan MongoDB sebagai sumber migrasi data homogen

- Anda dapat memigrasikan semua indeks sekunder yang didukung Amazon DocumentDB selama fase pemuatan penuh.
- AWS DMS memigrasikan koleksi secara paralel. migrasi data homogen menghitung segmen saat runtime berdasarkan ukuran rata-rata setiap dokumen dalam koleksi untuk kinerja maksimum.
- DMS dapat mereplikasi indeks sekunder yang Anda buat dalam fase CDC. DMS mendukung fitur ini di MongoDB versi 6.0.
- DMS mendukung dokumen dengan tingkat bersarang lebih besar dari 97.

Keterbatasan untuk menggunakan database yang kompatibel dengan MongoDB sebagai sumber migrasi data homogen

- Dokumen tidak dapat memiliki nama bidang dengan \$ awalan.
- AWS DMS tidak mendukung migrasi pengumpulan deret waktu.

- AWS DMS tidak mendukung `create`, `drop`, atau acara `rename collection` DDL selama fase CDC.
- AWS DMS tidak mendukung tipe data yang tidak konsisten dalam koleksi untuk bidang tersebut. `_id` Misalnya, koleksi yang tidak didukung berikut ini memiliki beberapa tipe data untuk `_id` bidang tersebut.

```
rs0 [direct: primary] test> db.collection1.aggregate([
...   {
...     $group: {
...       _id: { $type: "$_id" },
...       count: { $sum: 1 }
...     }
...   }
... ])
[ { _id: 'string', count: 6136 }, { _id: 'objectId', count: 848033 } ]
```

- Untuk tugas khusus CDC, AWS DMS hanya mendukung mode `immediate` mulai.
- AWS DMS tidak mendukung dokumen dengan karakter UTF8 yang tidak valid.
- AWS DMS tidak mendukung koleksi `sharded`.

Praktik terbaik untuk menggunakan database yang kompatibel dengan MongoDB sebagai sumber migrasi data homogen

- Untuk beberapa database besar dan koleksi yang dihosting pada instance MongoDB yang sama, kami sarankan Anda menggunakan aturan pemilihan untuk setiap database dan koleksi untuk membagi tugas antara beberapa tugas migrasi data dan proyek. Anda dapat menyetel basis data dan divisi koleksi Anda untuk kinerja maksimum.

Membuat penyedia data target untuk migrasi data homogen di AWS DMS

Anda dapat menggunakan database Amazon DocumentDB yang kompatibel dengan MySQL, PostgreSQL, dan Amazon DocumentDB sebagai penyedia data target untuk migrasi data homogen. AWS DMS

Untuk versi database yang didukung, lihat [Penyedia data target untuk migrasi data homogen DMS](#).

Penyedia data target Anda dapat berupa instans Amazon RDS DB atau kluster Amazon Aurora DB. Perhatikan bahwa versi database penyedia data target Anda harus sama atau lebih tinggi dari versi database penyedia data sumber Anda.

Topik

- [Menggunakan database MySQL yang kompatibel sebagai target untuk migrasi data homogen AWS DMS](#)
- [Menggunakan database PostgreSQL sebagai target untuk migrasi data homogen AWS DMS](#)
- [Menggunakan database Amazon DocumentDB sebagai target migrasi data homogen AWS DMS](#)

Menggunakan database MySQL yang kompatibel sebagai target untuk migrasi data homogen AWS DMS

Anda dapat menggunakan database MySQL yang kompatibel sebagai target migrasi untuk migrasi data homogen. AWS DMS

AWS DMS memerlukan izin tertentu untuk memigrasikan data ke database Amazon RDS for MySQL atau MariaDB atau Amazon Aurora MySQL target Anda. Gunakan skrip berikut untuk membuat pengguna database dengan izin yang diperlukan dalam database target MySQL Anda.

```
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'@'%';
```

Pada contoh sebelumnya, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri.

Gunakan skrip berikut untuk membuat pengguna database dengan izin yang diperlukan dalam database MariaDB Anda. Jalankan kueri GRANT untuk semua database tempat Anda bermigrasi.
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'@'%';
```

Pada contoh sebelumnya, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri.

Note

Di Amazon RDS, saat Anda mengaktifkan pencadangan otomatis untuk instance database MySQL/Maria, Anda juga mengaktifkan pencatatan biner. Ketika pengaturan ini diaktifkan, tugas migrasi data Anda mungkin gagal dengan kesalahan berikut saat membuat objek sekunder seperti fungsi, prosedur, dan pemicu pada database target. Jika database target Anda mengaktifkan pencatatan biner, maka setel `log_bin_trust_function_creators` ke `true` dalam grup parameter database sebelum memulai tugas.

```
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)
```

Keterbatasan untuk menggunakan database MySQL yang kompatibel sebagai target migrasi data homogen

Batasan berikut berlaku saat menggunakan database MySQL yang kompatibel sebagai target migrasi data homogen:

- Nama pengguna yang Anda gunakan untuk terhubung ke sumber data Anda memiliki batasan berikut:
 - Panjangnya bisa 2 hingga 64 karakter.
 - Tidak dapat memiliki spasi.
 - Dapat menyertakan karakter berikut: a-z, A-Z, 0-9, garis bawah (_).
 - Harus dimulai dengan a-z atau A-Z.
- Kata sandi yang Anda gunakan untuk menyambung ke sumber data Anda memiliki batasan berikut:
 - Panjangnya bisa 1 hingga 128 karakter.
 - Tidak dapat memuat salah satu dari berikut ini: kutipan tunggal ('), kutipan ganda ("), titik koma (;) atau spasi.

Menggunakan database PostgreSQL sebagai target untuk migrasi data homogen AWS DMS

Anda dapat menggunakan database PostgreSQL sebagai target migrasi untuk migrasi data homogen. AWS DMS

AWS DMS memerlukan izin tertentu untuk memigrasikan data ke database Amazon RDS for PostgreSQL atau Amazon Aurora PostgreSQL target Anda. Gunakan skrip berikut untuk membuat pengguna database dengan izin yang diperlukan dalam database target PostgreSQL Anda.

```
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;
      #For "Full load and change data capture (CDC)" and "Change data capture
      (CDC)" data migrations, setting up logical replication requires rds_superuser
      privileges
GRANT rds_superuser TO your_user;
```

Pada contoh sebelumnya, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri.

Untuk mengaktifkan replikasi logis untuk RDS Anda untuk target PostgreSQL, atur parameter dalam grup `rds.logical_replication` parameter DB Anda ke 1. Parameter statis ini memerlukan reboot dari instans DB atau cluster DB untuk diterapkan. Beberapa parameter bersifat statis, dan Anda hanya dapat mengaturnya di awal server. AWS DMS mengabaikan perubahan pada entri mereka di grup parameter DB sampai Anda me-restart server.

PostgreSQL menggunakan pemacu untuk mengimplementasikan kendala kunci asing. Selama fase beban penuh, AWS DMS muat setiap tabel satu per satu. Kami menyarankan Anda mematikan kendala kunci asing pada basis data target Anda selama pemuatan penuh. Untuk melakukannya, gunakan salah satu metode berikut.

- Matikan sementara semua pemacu untuk instance Anda, dan selesaikan beban penuh.
- Ubah nilai `session_replication_role` parameter di PostgreSQL.

Sewaktu-waktu, pemicu dapat berada di salah satu kondisi berikut: `origin`, `replica`, `always`, atau `disabled`. Saat Anda menyetel `session_replication_role` parameter ke `replica`, hanya pemicu dalam `replica` status yang aktif. Jika tidak, pemicu tetap tidak aktif.

Keterbatasan untuk menggunakan database yang kompatibel dengan PostgreSQL sebagai target migrasi data homogen

Batasan berikut berlaku saat menggunakan database yang kompatibel dengan PostgreSQL sebagai target migrasi data homogen:

- Nama pengguna yang Anda gunakan untuk terhubung ke sumber data Anda memiliki batasan berikut:
 - Panjangnya bisa 2 hingga 64 karakter.
 - Tidak dapat memiliki spasi.
 - Dapat menyertakan karakter berikut: a-z, A-Z, 0-9, garis bawah (`_`).
 - Harus dimulai dengan a-z atau A-Z.
- Kata sandi yang Anda gunakan untuk menyambung ke sumber data Anda memiliki batasan berikut:
 - Panjangnya bisa 1 hingga 128 karakter.
 - Tidak dapat memuat salah satu dari berikut ini: kutipan tunggal (`'`), kutipan ganda (`"`), titik koma (`;`) atau spasi.

Menggunakan database Amazon DocumentDB sebagai target migrasi data homogen AWS DMS

Anda dapat menggunakan database Amazon DocumentDB (dengan kompatibilitas MongoDB) dan cluster DocumentDB Elastic sebagai target migrasi untuk migrasi data homogen di AWS DMS

Untuk menjalankan migrasi data homogen untuk target Amazon DocumentDB, Anda dapat membuat akun pengguna dengan hak administrator, atau pengguna dengan izin baca/tulis hanya pada database untuk bermigrasi.

Migrasi data homogen mendukung semua tipe data BSON yang didukung Amazon DocumentDB. Untuk daftar tipe data ini, lihat [Jenis Data](#) di Panduan Pengembang Amazon DocumentDB.

Untuk menggunakan fitur pecahan cluster DocumentDB Elastic untuk memigrasikan koleksi non-sharded dari sumber, buat koleksi pecahan untuk dimigrasi sebelum memulai tugas migrasi data. Untuk informasi selengkapnya tentang pengumpulan pecahan di kluster Amazon DocumentDB Elastic, [lihat Langkah 5: Pecahkan](#) koleksi Anda di Panduan Pengembang Amazon DocumentDB.

Untuk target Amazon DocumentDB AWS DMS, mendukung mode atau SSLnone. require

Menjalankan migrasi data homogen di AWS DMS

Anda dapat menggunakannya AWS DMS untuk [Migrasi data homogen](#) memigrasikan data dari database sumber ke mesin yang setara di Amazon Relational Database Service (Amazon RDS), Amazon Aurora, atau Amazon DocumentDB. AWS DMS mengotomatiskan proses migrasi data dengan menggunakan alat database asli di basis data sumber dan target Anda.

Setelah Anda membuat profil instans dan penyedia data yang kompatibel untuk migrasi data homogen, buat proyek migrasi. Untuk informasi selengkapnya, lihat [Membuat proyek migrasi](#).

Bagian berikut menjelaskan cara membuat, mengkonfigurasi, dan menjalankan migrasi data yang homogen.

Topik

- [Membuat migrasi data di AWS DMS](#)
- [Aturan pemilihan untuk migrasi data homogen](#)
- [Mengelola migrasi data di AWS DMS](#)
- [Memantau migrasi data di AWS DMS](#)
- [Status migrasi data homogen di AWS DMS](#)
- [Migrasi data dari database MySQL dengan migrasi data homogen AWS DMS](#)
- [Migrasi data dari database PostgreSQL dengan migrasi data homogen di AWS DMS](#)
- [Migrasi data dari database MongoDB dengan migrasi data homogen di AWS DMS](#)

Membuat migrasi data di AWS DMS

Setelah Anda membuat proyek migrasi dengan penyedia data yang kompatibel dengan jenis yang sama, Anda dapat menggunakan proyek ini untuk migrasi data homogen. Untuk informasi selengkapnya, lihat [Membuat proyek migrasi](#).

Untuk mulai menggunakan migrasi data homogen, buat migrasi data baru. Anda dapat membuat beberapa migrasi data homogen dari berbagai jenis dalam satu proyek migrasi.

AWS DMS memiliki jumlah maksimum migrasi data homogen yang dapat Anda buat untuk Akun AWS. Lihat bagian berikut untuk informasi tentang kuota AWS DMS [Kuota untuk AWS Database Migration Service](#) layanan.

Sebelum membuat migrasi data, pastikan Anda menyiapkan sumber daya yang diperlukan seperti basis data sumber dan target, kebijakan dan peran IAM, profil instans, dan penyedia data. Lihat informasi selengkapnya di [Membuat sumber daya IAM](#), [Membuat profil instance](#), dan [Membuat penyedia data](#).

Selain itu, kami menyarankan agar Anda tidak menggunakan migrasi data homogen untuk memigrasikan data dari versi database yang lebih tinggi ke versi database yang lebih rendah. Periksa versi database yang Anda gunakan untuk penyedia data sumber dan target, dan tingkatkan versi basis data target Anda, jika diperlukan.

Untuk membuat migrasi data

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.
3. Pilih proyek migrasi Anda, dan pada tab Migrasi data, pilih Buat migrasi data.
4. Untuk Nama, masukkan nama untuk migrasi data Anda. Pastikan Anda menggunakan nama unik untuk migrasi data sehingga Anda dapat dengan mudah mengidentifikasinya.
5. Untuk tipe Replikasi, pilih jenis migrasi data yang ingin Anda konfigurasi. Anda dapat memilih salah satu opsi berikut.
 - Beban penuh — Memigrasi data sumber yang ada.
 - Pengambilan data beban dan perubahan penuh (CDC) — Memigrasi data sumber yang ada dan mereplikasi perubahan yang sedang berlangsung.
 - Ubah pengambilan data (CDC) - Mereplikasi perubahan yang sedang berlangsung.
6. Pilih kotak centang untuk Aktifkan CloudWatch log untuk menyimpan log migrasi data di Amazon CloudWatch. Jika Anda tidak memilih opsi ini, maka Anda tidak dapat melihat file log ketika migrasi data Anda gagal.
7. (Opsional) Perluas pengaturan lanjutan. Untuk Jumlah pekerjaan, masukkan jumlah thread paralel yang AWS DMS dapat digunakan untuk memigrasikan data sumber Anda ke target.

8. Untuk peran layanan IAM, pilih peran IAM yang Anda buat dalam prasyarat. Untuk informasi selengkapnya, lihat [Membuat peran IAM untuk migrasi data homogen di AWS DMS](#).
9. Konfigurasi mode Mulai untuk migrasi data tipe Change data capture (CDC). Anda dapat memilih salah satu opsi berikut.

- Segera — Memulai replikasi yang sedang berlangsung saat Anda memulai migrasi data.
- Menggunakan titik awal asli - Memulai replikasi yang sedang berlangsung dari titik yang ditentukan.

Untuk database PostgreSQL, masukkan nama slot replikasi logis untuk nama Slot dan masukkan nomor urutan log transaksi untuk titik awal asli.

Untuk database MySQL, masukkan nomor urut log transaksi untuk nomor urutan Log (LSN).

10. Konfigurasi mode Stop untuk migrasi data tipe Change data capture (CDC) atau Full load and change data capture (CDC). Anda dapat memilih salah satu opsi berikut.

- Jangan hentikan CDC — AWS DMS lanjutkan replikasi yang sedang berlangsung hingga Anda menghentikan migrasi data.
- Menggunakan titik waktu server - AWS DMS menghentikan replikasi yang sedang berlangsung pada waktu yang ditentukan.

Jika Anda memilih opsi ini, maka untuk tanggal dan waktu Berhenti, masukkan tanggal dan waktu ketika Anda ingin menghentikan replikasi yang sedang berlangsung secara otomatis.

11. Pilih Buat migrasi data.

AWS DMS membuat migrasi data Anda dan menambahkannya ke daftar di tab Migrasi data di proyek migrasi Anda. Di sini Anda dapat melihat status migrasi data Anda. Untuk informasi selengkapnya, lihat [Status migrasi](#).

 Important

Untuk migrasi data tipe Full load dan Full load and change data capture (CDC), AWS DMS hapus semua data, tabel, dan objek database lainnya pada database target Anda. Pastikan Anda memiliki cadangan database target Anda.

Setelah AWS DMS membuat migrasi data Anda, status migrasi data ini disetel ke Siap. Untuk memigrasikan data, Anda harus memulai migrasi data secara manual. Untuk melakukannya, pilih migrasi data Anda dari daftar. Selanjutnya, untuk Tindakan, pilih Mulai. Untuk informasi selengkapnya, lihat [Mengelola migrasi data](#).

Peluncuran pertama migrasi data homogen memerlukan beberapa pengaturan. AWS DMS menciptakan lingkungan tanpa server untuk migrasi data Anda. Proses ini memakan waktu hingga 15 menit. Setelah menghentikan dan memulai ulang migrasi data, AWS DMS tidak membuat lingkungan lagi, dan Anda dapat mengakses migrasi data lebih cepat.

Aturan pemilihan untuk migrasi data homogen

Anda dapat menggunakan aturan pemilihan untuk memilih skema, tabel, atau keduanya yang ingin Anda sertakan dalam replikasi Anda.

Note

AWS DMS hanya mendukung aturan pemilihan untuk migrasi data homogen saat menggunakan database yang kompatibel dengan MongoDB sebagai sumber.

Saat membuat tugas migrasi data, pilih Tambahkan aturan pemilihan.

Untuk pengaturan aturan, berikan nilai berikut:

- Skema: Pilih Masukkan skema.
- Nama skema: Berikan nama skema yang ingin Anda tiru, atau gunakan % sebagai wildcard.
- Nama tabel: Berikan nama tabel yang ingin Anda tiru, atau gunakan % sebagai wildcard.

Secara default, satu-satunya tindakan aturan yang didukung DMS adalah `Include`, dan satu-satunya karakter wildcard yang didukung DMS adalah `%`.

Example Memigrasi semua tabel dalam skema

Contoh berikut bermigrasi semua tabel dari skema bernama `dmsstd` di sumber Anda ke titik akhir target Anda.

```
{
  "rules": [
```

```

    {
      "rule-type": "selection",
      "rule-action": "include",
      "object-locator": {
        "schema-name": "dmsst",
        "table-name": "%"
      },
      "filters": [],
      "rule-id": "1",
      "rule-name": "1"
    }
  ]
}

```

Example Memigrasi beberapa tabel dalam skema

Contoh berikut memigrasikan semua tabel dengan nama yang dimulai dengan `collectionTest`, dari skema bernama `dmsst` di sumber Anda ke titik akhir target Anda.

```

{
  "rules": [
    {
      "rule-type": "selection",
      "rule-action": "include",
      "object-locator": {
        "schema-name": "dmsst",
        "table-name": "collectionTest%"
      },
      "filters": [],
      "rule-id": "1",
      "rule-name": "1"
    }
  ]
}

```

Example Migrasikan tabel tertentu dari beberapa skema

Contoh berikut memigrasikan beberapa tabel dari beberapa skema bernama `dmsst` dan `Test` di sumber Anda ke titik akhir target Anda.

```

{
  "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"
  }
]
```

Mengelola migrasi data di AWS DMS

Setelah Anda membuat migrasi data, AWS DMS tidak secara otomatis mulai memigrasi data. Anda memulai migrasi data secara manual bila diperlukan.

Sebelum memulai migrasi data, Anda dapat mengubah semua pengaturan migrasi data. Setelah memulai migrasi data, Anda tidak dapat mengubah jenis replikasi. Untuk menggunakan jenis replikasi lain, buat migrasi data baru.

Untuk memulai migrasi data

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.
3. Pilih proyek migrasi Anda. Pada tab Migrasi data, pilih migrasi data Anda. Halaman Ringkasan untuk migrasi data Anda terbuka.
4. Untuk Tindakan, pilih Mulai.

Setelah ini, AWS DMS buat lingkungan tanpa server untuk migrasi data Anda. Proses ini memakan waktu hingga 15 menit.

Setelah Anda memulai migrasi data, AWS DMS tetapkan statusnya ke Mulai. Status berikutnya yang AWS DMS digunakan untuk migrasi data Anda, bergantung pada jenis replikasi yang Anda pilih dalam setelan migrasi data. Untuk informasi selengkapnya, lihat [Status migrasi](#).

Untuk mengubah migrasi data

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.
3. Pilih proyek migrasi Anda. Pada tab Migrasi data, pilih migrasi data Anda. Halaman Ringkasan untuk migrasi data Anda terbuka.
4. Pilih Ubah.
5. Konfigurasi pengaturan untuk migrasi data Anda.

 Important

Jika Anda telah memulai migrasi data, maka Anda tidak dapat mengubah jenis replikasi.

6. Untuk melihat log migrasi data Anda di Amazon CloudWatch, pilih kotak centang untuk Aktifkan CloudWatch log.
7. Pilih Simpan perubahan.

Setelah AWS DMS memulai migrasi data, Anda dapat menghentikannya. Untuk melakukannya, pilih migrasi data Anda di tab Migrasi data. Selanjutnya, untuk Actions, pilih Stop.

Setelah Anda menghentikan migrasi data, AWS DMS tetapkan statusnya ke Berhenti. Selanjutnya, AWS DMS tetapkan status migrasi data ini ke Berhenti. Setelah AWS DMS menghentikan migrasi data, Anda dapat mengubah, melanjutkan, memulai ulang, atau menghapus migrasi data Anda.

Untuk melanjutkan replikasi data, pilih migrasi data yang Anda hentikan pada tab Migrasi data. Selanjutnya, untuk Tindakan, pilih Lanjutkan pemrosesan.

Untuk memulai ulang pemuatan data, pilih migrasi data yang Anda hentikan pada tab Migrasi data. Selanjutnya, untuk Actions, pilih Restart. AWS DMS menghapus semua data dari database target Anda dan memulai migrasi data dari awal.

Anda dapat menghapus migrasi data yang telah Anda hentikan atau yang belum Anda mulai. Untuk menghapus migrasi data, pilih di tab Migrasi data. Selanjutnya, untuk Tindakan, pilih Hapus. Untuk menghapus proyek migrasi Anda, hentikan dan hapus semua migrasi data.

Memantau migrasi data di AWS DMS

Setelah memulai migrasi data homogen, Anda dapat memantau status dan kemajuannya. Migrasi data dari kumpulan data besar seperti ratusan gigabyte membutuhkan waktu berjam-jam untuk diselesaikan. Untuk menjaga keandalan, ketersediaan, dan kinerja tinggi migrasi data Anda, pantau perkembangannya secara teratur.

Untuk memeriksa status dan kemajuan migrasi data Anda

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.
3. Pilih proyek migrasi Anda dan arahkan ke tab Migrasi data.
4. Untuk migrasi data Anda, lihat kolom Status. Untuk informasi selengkapnya tentang nilai di kolom ini, lihat [Status migrasi](#).
5. Untuk migrasi data yang sedang berjalan, kolom kemajuan Migrasi menampilkan persentase data yang dimigrasi.

Untuk memeriksa detail migrasi data Anda

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.
3. Pilih proyek migrasi Anda. Pada tab Migrasi data, pilih migrasi data Anda.
4. Pada tab Detail, Anda dapat melihat kemajuan migrasi. Khususnya, Anda dapat melihat metrik berikut.
 - Alamat IP Publik — Alamat IP publik dari migrasi data Anda. Anda memerlukan nilai ini untuk mengkonfigurasi jaringan. Untuk informasi selengkapnya, lihat [Menyiapkan jaringan](#).

- Tabel dimuat - Jumlah tabel yang berhasil dimuat.
 - Pemuatan tabel - Jumlah tabel yang sedang dimuat.
 - Tabel antrian — Jumlah tabel yang saat ini menunggu untuk dimuat.
 - Tabel errored — Jumlah tabel yang gagal dimuat.
 - Waktu berlalu — Jumlah waktu yang berlalu setelah dimulainya migrasi data Anda.
 - Latensi CDC — Waktu rata-rata yang berlalu antara saat perubahan terjadi pada tabel sumber dan saat AWS DMS menerapkan perubahan ini ke tabel target.
 - Migrasi dimulai — Waktu ketika Anda memulai migrasi data ini.
 - Migrasi berhenti — Waktu ketika Anda menghentikan migrasi data ini.
5. Untuk melihat file log untuk migrasi data Anda, pilih Lihat CloudWatch log di bawah Setelan migrasi data homogen. Anda dapat Mengaktifkan CloudWatch log saat membuat atau memodifikasi migrasi data. Lihat informasi yang lebih lengkap di [Membuat migrasi data](#) dan [Mengelola migrasi data](#).

Anda dapat menggunakan CloudWatch alarm atau peristiwa Amazon untuk melacak migrasi data Anda dengan cermat. Untuk informasi selengkapnya, lihat [Apa itu Amazon CloudWatch, CloudWatch Acara Amazon, dan CloudWatch Log Amazon?](#) di Panduan CloudWatch Pengguna Amazon. Perhatikan bahwa ada biaya untuk menggunakan Amazon CloudWatch.

Untuk migrasi data homogen, AWS DMS sertakan metrik berikut di Amazon. CloudWatch

Metrik	Deskripsi
OverallCDClatency	<p>Latensi keseluruhan selama fase CDC.</p> <p>Untuk database MySQL, metrik ini menunjukkan jumlah detik yang lewat antara perubahan dalam log biner sumber dan replikasi perubahan ini.</p> <p>Untuk database PostgreSQL, metrik ini menunjukkan jumlah detik yang lewat antara dan dari tampilan. <code>last_msg_receipt_time</code> <code>last_msg_send_time</code> <code>pg_stat_subscription</code></p> <p>Unit: detik</p>
StorageConsumption	Penyimpanan yang digunakan oleh migrasi data Anda.

Metrik	Deskripsi
	Satuan: Byte

Status migrasi data homogen di AWS DMS

Untuk setiap migrasi data yang Anda jalankan, AWS DMS menampilkan Status di AWS DMS konsol. Daftar berikut mencakup status yang tersedia.

- **Creating**— AWS DMS adalah membuat migrasi data.
- **Ready** Migrasi data sudah siap untuk dimulai.
- **Starting**— AWS DMS menciptakan lingkungan tanpa server untuk migrasi data Anda. Proses ini memakan waktu hingga 15 menit.
- **Load running**— AWS DMS sedang melakukan migrasi beban penuh.
- **Load complete, replication ongoing**— AWS DMS menyelesaikan beban penuh dan sekarang mereplikasi perubahan yang sedang berlangsung. AWS DMS menggunakan status ini hanya untuk migrasi data tipe full load dan change data capture (CDC).
- **Replication ongoing**- AWS DMS mereplikasi perubahan yang sedang berlangsung. AWS DMS menggunakan status ini hanya untuk migrasi tipe change data capture (CDC).
- **Reloading target**— AWS DMS memulai ulang migrasi data dan melakukan jenis migrasi yang ditentukan.
- **Stopping** Menghentikan migrasi data. AWS DMS menetapkan status ini setelah Anda memilih untuk menghentikan migrasi data pada menu Tindakan.
- **Stopped**— AWS DMS telah menghentikan migrasi data.
- **Failed** Migrasi data telah gagal. Untuk informasi selengkapnya, lihat file log.

Untuk melihat file log, pilih migrasi data Anda di tab Migrasi data. Selanjutnya, pilih Lihat CloudWatch log di bawah Setelan migrasi data homogen.

Important

Anda dapat melihat file log jika Anda memilih kotak centang untuk Aktifkan CloudWatch log saat Anda membuat migrasi data.

- **Deleting**— AWS DMS menghapus migrasi data. AWS DMS menetapkan status ini setelah Anda memilih untuk menghapus migrasi data pada menu Tindakan.

Migrasi data dari database MySQL dengan migrasi data homogen AWS DMS

Anda dapat menggunakan [Migrasi data homogen](#) untuk memigrasikan database MySQL yang dikelola sendiri ke RDS untuk MySQL atau Aurora MySQL. AWS DMS menciptakan lingkungan tanpa server untuk migrasi data Anda. Untuk berbagai jenis migrasi data, AWS DMS gunakan alat database MySQL asli yang berbeda.

Untuk migrasi data homogen dari tipe Full load, AWS DMS gunakan `mysqldumper` untuk membaca data dari database sumber Anda dan menyimpannya di disk yang terpasang ke lingkungan tanpa server. Setelah AWS DMS membaca semua data sumber Anda, ia menggunakan `mysqlloader` di database target untuk memulihkan data Anda.

Untuk migrasi data homogen tipe Full load and change data capture (CDC), AWS DMS gunakan `mysqldumper` untuk membaca data dari database sumber Anda dan menyimpannya di disk yang terpasang ke lingkungan tanpa server. Setelah AWS DMS membaca semua data sumber Anda, ia menggunakan `mysqlloader` di database target untuk memulihkan data Anda. Setelah AWS DMS menyelesaikan beban penuh, ia mengatur replikasi binlog dengan posisi binlog diatur ke awal beban penuh. Untuk menghindari inkonsistensi data, atur Jumlah pekerjaan ke 1 untuk menangkap status konsisten data yang ada. Untuk informasi selengkapnya, lihat [Membuat migrasi data](#).

Untuk migrasi data homogen tipe Change data capture (CDC), AWS DMS memerlukan titik awal CDC Asli untuk memulai replikasi. Jika Anda memberikan titik awal CDC asli, maka AWS DMS menangkap perubahan dari titik itu. Atau, pilih Segera di pengaturan migrasi data untuk secara otomatis menangkap titik awal replikasi saat migrasi data aktual dimulai.

Note

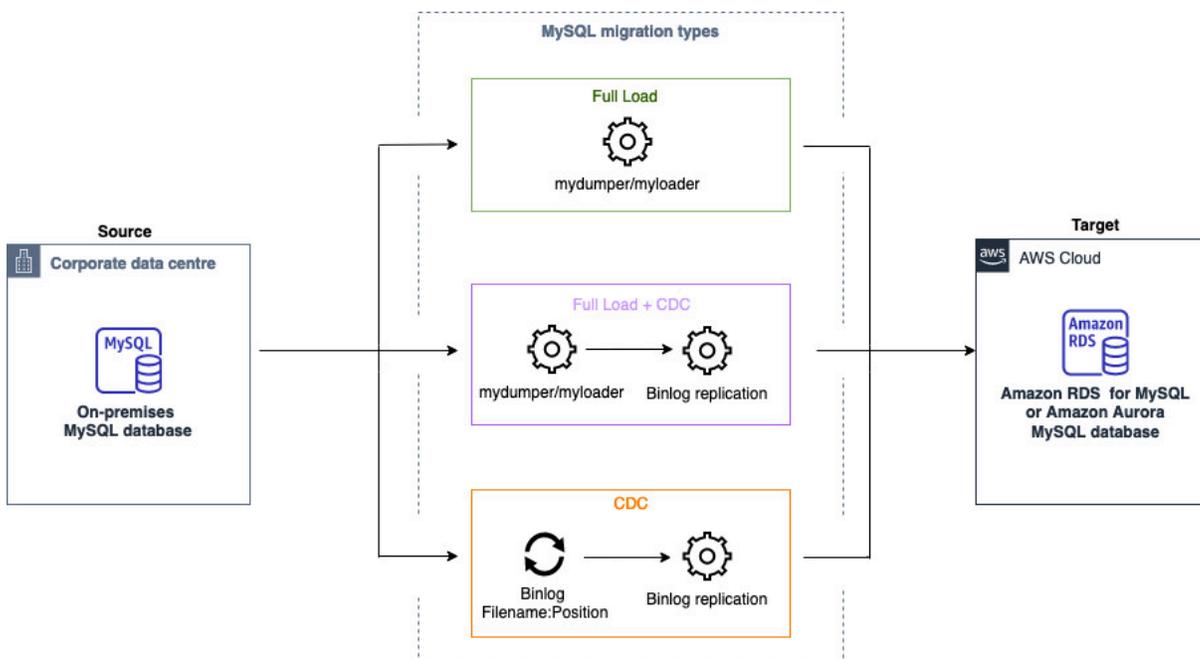
Agar migrasi khusus CDC berfungsi dengan baik, semua skema dan objek basis data sumber harus sudah ada di database target. Target mungkin memiliki objek yang tidak ada pada sumbernya.

Anda dapat menggunakan contoh kode berikut untuk mendapatkan nomor urutan log saat ini (LSN) di database MySQL Anda.

```
show master status
```

Query ini mengembalikan nama file binlog dan posisi. Untuk titik awal asli, gunakan kombinasi nama file binlog dan posisi. Misalnya, `mysql-bin-changelog.000024:373`. Dalam contoh ini, `mysql-bin-changelog.000024` adalah nama file binlog dan 373 merupakan posisi di mana AWS DMS mulai menangkap perubahan.

Diagram berikut menunjukkan proses menggunakan migrasi data homogen untuk memigrasikan database MySQL AWS DMS ke RDS untuk MySQL atau Aurora MySQL.



Migrasi data dari database PostgreSQL dengan migrasi data homogen di AWS DMS

Anda dapat menggunakan [Migrasi data homogen](#) untuk memigrasikan database PostgreSQL yang dikelola sendiri ke RDS untuk PostgreSQL atau Aurora PostgreSQL. AWS DMS menciptakan lingkungan tanpa server untuk migrasi data Anda. Untuk berbagai jenis migrasi data, AWS DMS gunakan alat database PostgreSQL asli yang berbeda.

Untuk migrasi data homogen dari tipe Full load, AWS DMS gunakan `pg_dump` untuk membaca data dari database sumber Anda dan menyimpannya di disk yang terpasang ke lingkungan tanpa server. Setelah AWS DMS membaca semua data sumber Anda, ia menggunakan `pg_restore` di database target untuk memulihkan data Anda.

Untuk migrasi data homogen tipe Full load and change data capture (CDC), AWS DMS gunakan `pg_dump` untuk membaca data dari database sumber Anda dan menyimpannya di disk yang terpasang ke lingkungan tanpa server. Setelah AWS DMS membaca semua data sumber Anda, ia menggunakan `pg_restore` di database target untuk memulihkan data Anda. Setelah AWS DMS menyelesaikan beban penuh, secara otomatis beralih ke model penerbit dan pelanggan untuk replikasi logis. Dalam model ini, satu atau lebih pelanggan berlangganan satu atau lebih publikasi pada node penerbit.

Untuk migrasi data homogen tipe Change data capture (CDC), AWS DMS memerlukan titik awal asli untuk memulai replikasi. Jika Anda memberikan titik awal asli, maka AWS DMS menangkap perubahan dari titik itu. Atau, pilih Segera di pengaturan migrasi data untuk secara otomatis menangkap titik awal replikasi saat migrasi data aktual dimulai.

Note

Agar migrasi khusus CDC berfungsi dengan baik, semua skema dan objek basis data sumber harus sudah ada di database target. Target mungkin memiliki objek yang tidak ada pada sumbernya.

Anda dapat menggunakan contoh kode berikut untuk mendapatkan titik awal asli dalam database PostgreSQL Anda.

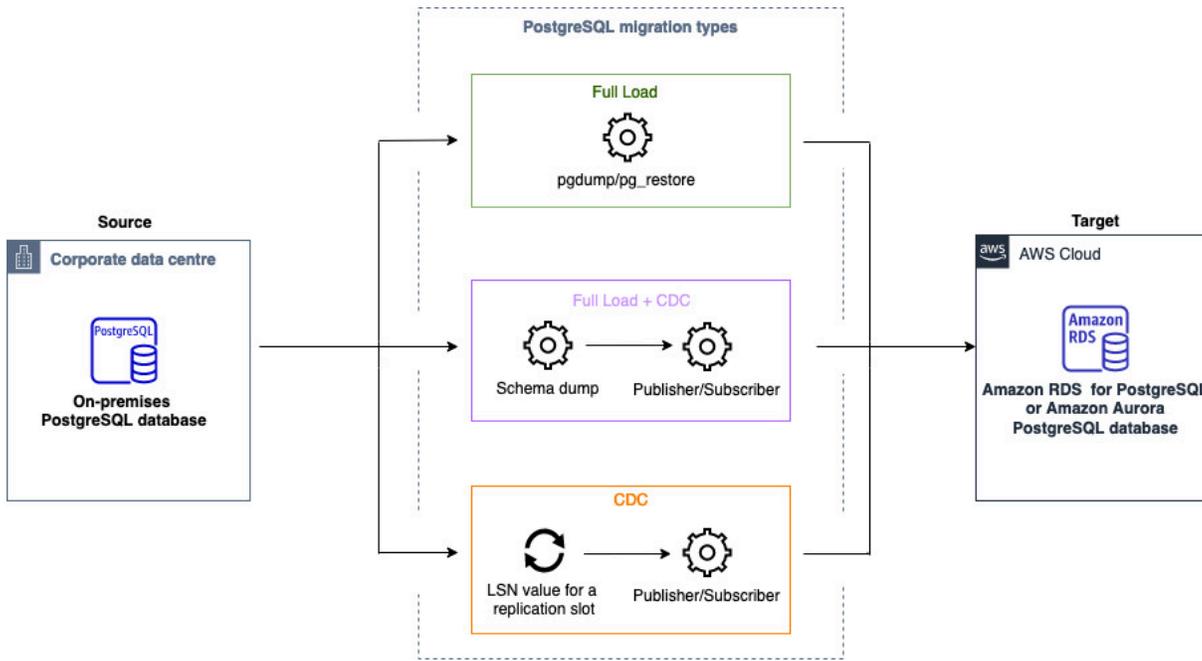
```
select confirmed_flush_lsn from pg_replication_slots where
slot_name='migrate_to_target';
```

Kueri ini menggunakan `pg_replication_slots` tampilan dalam database PostgreSQL Anda untuk menangkap nilai nomor urutan log (LSN).

Setelah AWS DMS menetapkan status migrasi data homogen PostgreSQL Anda ke Dihentikan, Gagal, atau Dihapus, penerbit dan replikasi tidak akan dihapus. Jika Anda tidak ingin melanjutkan migrasi, hapus slot replikasi dan penerbit dengan menggunakan perintah berikut.

```
SELECT pg_drop_replication_slot('migration_subscriber_{ARN}');
DROP PUBLICATION publication_{ARN};
```

Diagram berikut menunjukkan proses menggunakan migrasi data homogen untuk memigrasikan database PostgreSQL AWS DMS ke RDS untuk PostgreSQL atau Aurora PostgreSQL.



Migrasi data dari database MongoDB dengan migrasi data homogen di AWS DMS

Anda dapat menggunakan [Migrasi data homogen](#) untuk memigrasikan database MongoDB yang dikelola sendiri ke Amazon DocumentDB. AWS DMS menciptakan lingkungan tanpa server untuk migrasi data Anda. Untuk berbagai jenis migrasi data, AWS DMS gunakan alat database MongoDB asli yang berbeda.

Untuk migrasi data homogen dari tipe Full load, AWS DMS gunakan mongodump untuk membaca data dari database sumber Anda dan menyimpannya di disk yang terpasang ke lingkungan tanpa server. Setelah AWS DMS membaca semua data sumber Anda, ia menggunakan mongorestore dalam database target untuk memulihkan data Anda.

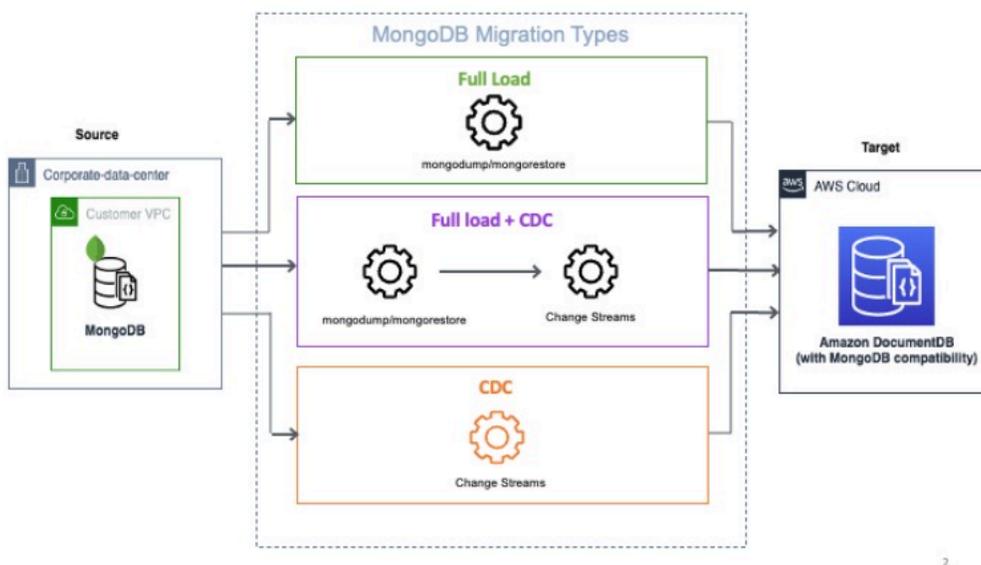
Untuk migrasi data homogen dari tipe Full load and change data capture (CDC), AWS DMS gunakan mongodump untuk membaca data dari database sumber Anda dan menyimpannya di disk yang terpasang ke lingkungan tanpa server. Setelah AWS DMS membaca semua data sumber Anda, ia menggunakan mongorestore dalam database target untuk memulihkan data Anda. Setelah AWS DMS menyelesaikan beban penuh, secara otomatis beralih ke model penerbit dan pelanggan untuk replikasi logis. Dalam model ini, kami merekomendasikan ukuran oplog untuk mempertahankan perubahan setidaknya selama 24 jam.

Untuk migrasi data homogen dari jenis Ubah pengambilan data (CDC), pilih `immediately` di pengaturan migrasi data untuk secara otomatis menangkap titik awal replikasi saat migrasi data aktual dimulai.

Note

Untuk koleksi baru atau berganti nama, Anda perlu membuat tugas migrasi data baru untuk koleksi tersebut sebagai migrasi data homogen. Untuk sumber yang kompatibel dengan MongoDB, AWS DMS tidak mendukung `create`, `rename` dan operasi `drop collection`

Diagram berikut menunjukkan proses menggunakan migrasi data homogen untuk memigrasikan database MongoDB AWS DMS ke Amazon DocumentDB.



Pemecahan masalah untuk migrasi data homogen diAWS DMS

Dalam daftar berikut, Anda dapat menemukan tindakan yang harus diambil saat Anda mengalami masalah dengan migrasi data homogen diAWS DMS.

Topik

- [Saya tidak dapat membuat migrasi data homogen diAWS DMS](#)
- [Saya tidak dapat memulai migrasi data homogen diAWS DMS](#)
- [Saya tidak dapat terhubung ke database target saat menjalankan migrasi data diAWS DMS](#)
- [AWS DMSmemigrasikan tampilan sebagai tabel di PostgreSQL](#)

Saya tidak dapat membuat migrasi data homogen diAWS DMS

Jika Anda mendapatkan pesan kesalahan yang mengatakan bahwaAWS DMStidak dapat terhubung ke penyedia data Anda setelah Anda memilihBuat migrasi data, kemudian pastikan bahwa Anda telah mengkonfigurasi peran IAM yang diperlukan. Untuk informasi selengkapnya, lihat [Membuat peran IAM](#).

Jika Anda telah mengkonfigurasi peran IAM dan masih mendapatkan pesan galat ini, maka tambahkan peran IAM ini ke pengguna kunci Anda diAWS KMSkonfigurasi kunci. Untuk informasi lebih lanjut, lihat[Memungkinkan pengguna kunci untuk menggunakan kunci KMS](#)di dalamAWS Key Management ServicePanduan Pengembang.

Saya tidak dapat memulai migrasi data homogen diAWS DMS

Jika Anda mendapatkanFailedstatus saat Anda memulai migrasi data di proyek migrasi, periksa versi penyedia data sumber dan target Anda. Untuk melakukannya, jalankanSELECT VERSION() ;kueri di database MySQL atau PostgreSQL Anda. Pastikan Anda menggunakan versi database yang didukung.

Untuk daftar database sumber yang didukung, lihat[Sumber untuk migrasi data homogen DMS](#).

Untuk daftar database target yang didukung, lihat[Target untuk migrasi data homogen DMS](#).

Jika Anda menggunakan versi database yang tidak didukung, maka tingkatkan database sumber atau target Anda, dan coba lagi.

Periksa pesan galat untuk migrasi data Anda diAWS DMSkonsol. Untuk melakukannya, buka proyek migrasi Anda, dan pilih migrasi data Anda. PadaDetailtab, periksaPesan kegagalan terakhirdi bawahUmum.

Akhirnya, analisisCloudWatchlog. Untuk melakukannya, buka proyek migrasi Anda, dan pilih migrasi data Anda. PadaDetailtab, pilihLihatCloudWatchlog.

Saya tidak dapat terhubung ke database target saat menjalankan migrasi data diAWS DMS

Jika Anda mendapatkanTidak dapat menyambung ke targetpesan kesalahan, kemudian melakukan tindakan berikut.

1. Pastikan grup keamanan yang dilampirkan ke database sumber dan target Anda berisi aturan untuk lalu lintas masuk dan keluar apa pun. Untuk informasi selengkapnya, lihat [Mengkonfigurasi replikasi data yang sedang berlangsung](#).
2. Verifikasi daftar kontrol akses jaringan (ACL) dan aturan tabel rute.
3. Database Anda harus dapat diakses dari VPC yang Anda buat. Tambahkan alamat IP publik di grup keamanan VPC, dan izinkan koneksi input di firewall Anda.
4. Pada Migrasi data proyek migrasi Anda, pilih migrasi data Anda. Catat alamat IP publik di bawah Konektivitas dan keamanan pada Detail tab. Selanjutnya, izinkan akses dari alamat IP publik migrasi data Anda di database sumber dan target Anda.
5. Untuk replikasi data yang sedang berlangsung, pastikan database sumber dan target Anda dapat berkomunikasi satu sama lain.

Untuk informasi lebih lanjut, lihat [Mengontrol lalu lintas ke sumber daya menggunakan grup keamanan](#) di dalam Panduan Pengguna Cloud Pribadi Virtual Amazon.

AWS DMS memigrasikan tampilan sebagai tabel di PostgreSQL

Migrasi data homogen tidak mendukung migrasi tampilan sebagai tampilan di PostgreSQL. Untuk PostgreSQL, AWS DMS bermigrasi pandangan sebagai tabel.

Bekerja dengan penyedia data, profil instans, dan proyek migrasi di AWS DMS

Saat Anda menggunakan Konversi Skema DMS dan migrasi data homogen di AWS Database Migration Service, Anda bekerja dengan proyek migrasi. Pada gilirannya, proyek AWS DMS migrasi menggunakan grup subnet, profil instance, dan penyedia data.

Sebuah subnet adalah rentang alamat IP di VPC Anda. Grup subnet replikasi menyertakan subnet dari Availability Zone yang berbeda yang dapat digunakan oleh profil instans Anda. Perhatikan bahwa grup subnet replikasi adalah sumber daya DMS, dan berbeda dari grup subnet yang digunakan Amazon VPC dan Amazon RDS.

Profil instans menentukan pengaturan jaringan dan keamanan untuk lingkungan tanpa server tempat proyek migrasi Anda berjalan.

Penyedia data menyimpan jenis penyimpanan data dan informasi lokasi tentang database Anda. Setelah menambahkan penyedia data ke proyek migrasi, Anda akan memberikan kredensi database. AWS Secrets Manager AWS DMS menggunakan informasi ini untuk terhubung ke database Anda.

Setelah membuat penyedia data, profil instans, dan AWS sumber daya lainnya, Anda dapat membuat proyek migrasi. Proyek migrasi menjelaskan profil instans Anda, penyedia data sumber dan target, dan rahasia dari AWS Secrets Manager. Anda dapat membuat beberapa proyek migrasi untuk penyedia data sumber dan target yang berbeda.

Anda melakukan sebagian besar pekerjaan Anda dalam proyek migrasi. Untuk Konversi Skema DMS, Anda menggunakan proyek migrasi untuk menilai objek penyedia data sumber Anda dan mengonversinya ke format yang kompatibel dengan database target. Kemudian, Anda dapat menerapkan kode yang dikonversi ke penyedia data target Anda atau menyimpannya sebagai skrip SQL. Untuk migrasi data homogen, Anda menggunakan proyek migrasi untuk memigrasikan data dari database sumber Anda ke database target dari jenis yang sama di database. AWS Cloud

Proyek migrasi AWS DMS hanya tanpa server. AWS DMS secara otomatis menyediakan sumber daya cloud untuk proyek migrasi Anda.

AWS DMS memiliki jumlah maksimum profil instans, penyedia data, dan proyek migrasi yang dapat Anda buat untuk Anda Akun AWS. Lihat bagian berikut untuk informasi tentang kuota AWS DMS [Kuota untuk AWS Database Migration Service](#) layanan.

Topik

- [Membuat grup subnet untuk proyek AWS DMS migrasi](#)
- [Membuat profil instance untuk AWS Database Migration Service](#)
- [Membuat penyedia data di AWS Database Migration Service](#)
- [Membuat proyek migrasi di AWS Database Migration Service](#)
- [Mengelola proyek migrasi di AWS Database Migration Service](#)

Membuat grup subnet untuk proyek AWS DMS migrasi

Sebelum Anda membuat profil instance, konfigurasi grup subnet untuk profil instans Anda.

Cara membuat grup subnet

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Di panel navigasi, pilih Grup subnet, lalu pilih Buat grup subnet.
3. Untuk Nama, masukkan nama unik grup subnet Anda.
4. Untuk Deskripsi, masukkan deskripsi singkat tentang grup subnet Anda.
5. Untuk VPC, pilih VPC yang memiliki setidaknya satu subnet di setidaknya dua Availability Zone.
6. Untuk Tambahkan subnet, pilih subnet untuk disertakan dalam grup subnet. Anda harus memilih subnet dalam setidaknya dua Availability Zone.

Untuk terhubung ke database Amazon RDS, tambahkan subnet publik ke grup subnet Anda.
Untuk terhubung ke database lokal, tambahkan subnet pribadi ke grup subnet Anda.

7. Pilih Buat grup subnet.

Membuat profil instance untuk AWS Database Migration Service

Anda dapat membuat beberapa profil instans di AWS DMS konsol. Pastikan Anda memilih profil instance yang akan digunakan untuk setiap proyek migrasi yang Anda buat AWS DMS.

Buat profil instans

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Di panel navigasi, pilih Profil instans.

3. Pilih Buat profil contoh.
4. Pada halaman Buat profil instance, masukkan nilai deskriptif untuk Nama untuk profil instans Anda.
5. Untuk tipe Jaringan, pilih mode Dual-stack untuk membuat profil instance yang mendukung pengalamatan IPv4 dan IPv6. Simpan opsi default untuk membuat profil instance yang hanya mendukung pengalamatan IPv4.
6. Selanjutnya, pilih Virtual private cloud (VPC) untuk menjalankan instance Anda dari jenis jaringan yang dipilih. Kemudian pilih grup Subnet dan grup keamanan VPC untuk profil instans Anda.

Untuk terhubung ke database Amazon RDS, gunakan grup subnet yang menyertakan subnet publik. Untuk terhubung ke database lokal, gunakan grup subnet yang menyertakan subnet pribadi. Pastikan Anda mengonfigurasi jaringan sehingga AWS DMS dapat mengakses basis data lokal sumber menggunakan alamat IP publik gateway NAT. Untuk informasi selengkapnya, lihat [Buat VPC berdasarkan Amazon VPC](#).

7. (Opsional) Jika Anda membuat proyek migrasi untuk Konversi Skema DMS, lalu untuk setelan konversi Skema - opsional, pilih bucket Amazon S3 untuk menyimpan informasi dari proyek migrasi Anda. Kemudian pilih peran AWS Identity and Access Management (IAM) yang menyediakan akses ke bucket Amazon S3 ini. Untuk informasi selengkapnya, lihat [Buat bucket Amazon S3](#).
8. Pilih Buat profil contoh.

Setelah membuat profil instans, Anda dapat memodifikasi atau menghapusnya.

Untuk memodifikasi profil instance

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih profil Instance. Halaman profil Instans terbuka.
3. Pilih profil instans Anda, lalu pilih Ubah.
4. Perbarui nama profil instans Anda, edit pengaturan bucket VPC atau Amazon S3.
5. Pilih Simpan perubahan.

Untuk menghapus profil instance

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih profil Instance. Halaman profil Instans terbuka.
3. Pilih profil instans Anda, lalu pilih Hapus.
4. Pilih Hapus untuk mengonfirmasi pilihan Anda.

Membuat penyedia data di AWS Database Migration Service

Anda dapat membuat penyedia data dan menggunakannya dalam proyek AWS DMS migrasi. Penyedia data Anda dapat berupa mesin yang dikelola sendiri yang berjalan di lokasi atau di instans Amazon EC2. Selain itu, penyedia data Anda dapat menjadi mesin yang dikelola sepenuhnya, seperti Amazon Relational Database Service (Amazon RDS) atau Amazon Aurora.

Untuk setiap database, Anda dapat membuat penyedia data tunggal. Anda dapat menggunakan penyedia data tunggal dalam beberapa proyek migrasi.

Sebelum membuat proyek migrasi, pastikan Anda telah membuat setidaknya dua penyedia data. Salah satu penyedia data Anda harus berada di Layanan AWS. Anda tidak dapat menggunakan AWS DMS untuk mengonversi skema atau memigrasikan data ke database lokal.

Prosedur berikut menunjukkan cara membuat penyedia data di wizard AWS DMS konsol.

Untuk membuat penyedia data

1. Masuk ke AWS Management Console, lalu buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih Penyedia data. Halaman penyedia data terbuka.
3. Pilih Buat penyedia data. Tabel berikut menjelaskan pengaturan.

Opsi	Tindakan
Konfigurasi	Pilih apakah akan memasukkan informasi tentang penyedia data Anda secara manual atau menggunakan instans Amazon RDS DB.

Opsi	Tindakan
Nama	Masukkan nama untuk penyedia data Anda. Pastikan Anda menggunakan nama unik untuk penyedia data Anda sehingga Anda dapat dengan mudah mengidentifikasinya.
Jenis mesin	Pilih jenis mesin database untuk penyedia data Anda.
Nama server	Masukkan nama Domain Name Service (DNS) atau alamat IP server database Anda. Nama server untuk penyedia data yang digunakan untuk replikasi homogen harus dimulai dengan karakter alfanumerik, dan hanya dapat berisi karakter alfanumerik, tanda hubung (-), periode (.), atau garis bawah (_).
Port	Masukkan port yang digunakan untuk terhubung ke server database Anda.
ID Layanan (SID) atau nama layanan	Masukkan ID Sistem Oracle (SID). Untuk menemukan SID Oracle, kirimkan kueri berikut ke database Oracle Anda: <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>SELECT sys_context('userenv','instance_name') AS SID FROM dual;</pre> </div>
Nama basis data	Masukkan nama database untuk penyedia data ini. Nama database untuk penyedia data yang digunakan untuk replikasi homogen dapat mencapai 63 karakter dan tidak dapat berisi spasi.
Mode Secure Socket Layer (SSL)	Pilih mode SSL jika Anda ingin mengaktifkan enkripsi koneksi untuk penyedia data ini. Bergantung pada mode yang Anda pilih, Anda mungkin perlu memberikan informasi sertifikat dan sertifikat server. Untuk detail lebih lanjut, lihat Menggunakan SSL dengan AWS Database Migration Service .
Modus otentikasi	Untuk sumber MongoDB, mode otentikasi yang digunakan untuk mengautentikasi AWS DMS koneksi endpoint.

Opsi	Tindakan
Sumber otentikasi	Untuk sumber MongoDB, nama database MongoDB yang akan digunakan untuk memvalidasi kredensi Anda untuk otentikasi.
Mekanisme otentikasi	Untuk sumber MongoDB, metode otentikasi yang digunakan MongoDB untuk mengenkripsi kata sandi.

4. Pilih Buat penyedia data.

Setelah Anda membuat penyedia data, pastikan bahwa Anda menambahkan kredensial koneksi database di. AWS Secrets Manager

Membuat proyek migrasi di AWS Database Migration Service

Sebelum membuat proyek migrasi di AWS DMS, pastikan Anda membuat sumber daya berikut:

- Penyedia data yang menggambarkan basis data sumber dan target Anda
- Rahasia dengan kredensial database yang disimpan di AWS Secrets Manager
- Peran AWS Identity and Access Management (IAM) yang menyediakan akses ke Secrets Manager
- Profil instance yang mencakup pengaturan jaringan dan keamanan

Untuk membuat proyek migrasi

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.
3. Pilih Buat proyek migrasi. Tabel berikut menjelaskan pengaturan.

Opsi	Tindakan
Nama	Masukkan nama untuk proyek migrasi Anda. Pastikan Anda menggunakan nama unik untuk proyek migrasi sehingga Anda dapat dengan mudah mengidentifikasinya.

Opsi	Tindakan
Profil contoh	Pilih profil instans yang akan digunakan untuk proyek migrasi Anda.
Sumber	Pilih Browse, lalu pilih penyedia data sumber Anda.
ID Rahasia	Pilih Amazon Resource Name (ARN) rahasia Anda di Secrets Manager yang menyimpan kredensi basis data sumber Anda.
Peran IAM	Pilih peran IAM untuk menyediakan akses ke kredensial basis data sumber Anda di Secrets Manager.
Target	Pilih Browse, lalu pilih penyedia data sumber Anda.
ID Rahasia	Pilih ARN rahasia Anda di Secrets Manager yang menyimpan kredensial basis data target Anda.
Peran IAM	Pilih peran IAM untuk menyediakan akses ke kredensial basis data target Anda di Secrets Manager.
Aturan transformasi	(Opsional) Jika Anda membuat proyek migrasi untuk Konversi Skema DMS, pilih Tambahkan aturan transformasi untuk mengatur aturan transformasi. Aturan transformasi memungkinkan Anda untuk mengubah nama objek sesuai dengan aturan yang Anda tentukan. Untuk informasi selengkapnya, lihat Menyiapkan aturan transformasi .

4. Pilih Buat proyek migrasi.

Setelah AWS DMS membuat proyek migrasi, Anda dapat menggunakan proyek ini di Konversi Skema DMS atau migrasi data homogen. Untuk mulai bekerja dengan proyek migrasi Anda, pada halaman Proyek migrasi, pilih proyek Anda dari daftar.

Mengelola proyek migrasi di AWS Database Migration Service

Setelah membuat proyek migrasi, Anda dapat memodifikasi atau menghapusnya. Misalnya, untuk mengubah penyedia data sumber atau target, ubah proyek migrasi Anda.

Anda dapat mengubah atau menghapus proyek migrasi hanya setelah Anda menutup konversi skema atau operasi migrasi data. Untuk melakukannya, pilih proyek migrasi Anda dari daftar, lalu pilih Konversi skema atau Migrasi data. Selanjutnya, pilih Tutup konversi skema untuk Konversi Skema DMS dan konfirmasi pilihan Anda. Untuk migrasi data homogen, pilih migrasi data Anda, lalu pilih Berhenti pada menu Tindakan. Setelah mengedit proyek migrasi, Anda dapat meluncurkan konversi skema atau memulai migrasi data lagi.

Untuk memodifikasi proyek migrasi

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.
3. Pilih proyek migrasi Anda, lalu pilih Ubah.
4. Perbarui nama proyek Anda, edit profil instans, atau ubah penyedia data sumber dan target. Secara opsional, tambahkan atau edit aturan migrasi yang mengubah nama objek selama konversi.
5. Pilih Simpan perubahan.

Untuk menghapus proyek migrasi

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih proyek Migrasi. Halaman proyek Migrasi terbuka.
3. Pilih proyek migrasi Anda, lalu pilih Hapus.
4. Pilih Hapus untuk mengonfirmasi pilihan Anda.

Praktik terbaik AWS Database Migration Service

Untuk menggunakan AWS Database Migration Service (AWS DMS) paling efektif, lihat rekomendasi bagian ini tentang cara paling efisien untuk melakukan migrasi data Anda.

Topik

- [Perencanaan migrasi untuk AWS Database Migration Service](#)
- [Mengubah skema](#)
- [Meninjau dokumentasi publik AWS DMS](#)
- [Menjalankan bukti konsep](#)
- [Meningkatkan performa suatu migrasi AWS DMS](#)
- [Menggunakan server nama on-premise Anda sendiri](#)
- [Melakukan migrasi objek biner large \(LOB\)](#)
- [Meningkatkan performa ketika melakukan migrasi tabel-tabel large menggunakan filter baris](#)
- [Replikasi yang sedang berlangsung](#)
- [Mengurangi beban pada basis data sumber Anda](#)
- [Mengurangi kemacetan pada basis data target Anda](#)
- [Menggunakan validasi data selama migrasi](#)
- [Pemantauan tugas AWS DMS menggunakan metrik](#)
- [Tindakan dan notifikasi](#)
- [Menggunakan cara dengan mencatat tugas untuk memecahkan masalah migrasi](#)
- [Memecahkan masalah tugas replikasi dengan Time Travel](#)
- [Mengubah pengguna dan skema untuk target Oracle](#)
- [Mengubah tempat penyimpanan tabel dan indeks untuk target Oracle](#)
- [Pemutakhiran versi instans replikasi](#)
- [Memahami biaya migrasi Anda](#)

Perencanaan migrasi untuk AWS Database Migration Service

Ketika merencanakan migrasi basis data menggunakan AWS Database Migration Service, pertimbangkan hal berikut:

- Untuk Connect pada sumber dan target basis data Anda ke instans replikasi AWS DMS, lakukan konfigurasi jaringan. Melakukan hal ini bisa sesederhana menghubungkan dua sumber daya AWS virtual private cloud (VPC) sebagai instans replikasi Anda. Ini dapat berkisar hingga konfigurasi yang lebih kompleks seperti menghubungkan basis data On-Premise ke instans DB Amazon RDS melalui Jaringan Pribadi Virtual (VPN). Untuk informasi selengkapnya, lihat [Konfigurasi jaringan untuk migrasi basis data](#).
- Titik akhir sumber dan target — Pastikan bahwa Anda mengetahui informasi apa dan tabel dalam sumber basis data yang perlu dilakukan migrasi ke basis data target. Migrasi skema Support Basic AWS DMS, termasuk penciptaan tabel dan kunci primer. Namun, AWS DMS tidak secara otomatis membuat indeks sekunder, kunci asing, akun pengguna, dan sebagainya, dalam basis data target. Tergantung pada mesin basis data sumber dan target Anda, Anda mungkin perlu pengaturan catatan tambahan atau memodifikasi pengaturan lain untuk basis data sumber atau target. Untuk informasi selengkapnya, lihat [Sumber untuk migrasi data](#) dan [Target migrasi data](#).
- Skema dan migrasi kode — AWS DMS tidak melakukan konversi skema atau kode. Anda dapat menggunakan perangkat seperti Oracle SQL Developer, MySQL Workbench, dan pgAdmin III untuk mengkonversi skema Anda. Untuk mengkonversi skema yang ada untuk mesin basis data yang berbeda, Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT). Cara ini dapat membuat skema target dan dapat menghasilkan dan membuat seluruh skema: tabel, indeks, tampilan, dan sebagainya. Anda juga dapat menggunakan perangkat untuk dapat mengubah PL/SQL atau TSQL ke PgSQL dan beberapa format lainnya. Untuk informasi selengkapnya tentang AWS SCT, lihat [Panduan Pengguna AWS SCT](#).
- Jenis data yang tidak didukung — Pastikan bahwa Anda dapat mengubah tipe data sumber ke tipe data yang setara untuk basis data target. Untuk informasi selengkapnya tentang jenis data yang didukung, lihat bagian sumber atau target untuk menyimpan data Anda.
- Hasil penulisan Support diagnostik — Saat merencanakan migrasi, sebaiknya jalankan penulisan dukungan diagnostik. Dengan hasil dari penulisan ini, Anda dapat menemukan informasi lanjutan tentang kemungkinan kegagalan migrasi.

Jika penulisan Support tersedia untuk basis data Anda, untuk mengunduh dapat menggunakan tautan dalam topik penulisan yang sesuai di bagian berikut. Setelah memeriksa dan meninjau penulisan ini, Anda dapat menjalankannya sesuai dengan prosedur yang dijelaskan dalam topik penulisan di lingkungan setempat Anda. Ketika selesai menjalankan penulisan, Anda dapat meninjau hasilnya. Kami menyarankan agar menjalankan penulisan ini sebagai langkah pertama dari setiap upaya pemecahan masalah. Hasilnya dapat berguna saat bekerja dengan Tim AWS Support. Untuk informasi selengkapnya, lihat [Bekerja dengan skrip dukungan diagnostik di AWS DMS](#).

- **Penilaian Pra-migrasi** — Penilaian pra-migrasi mengevaluasi komponen tertentu dari tugas migrasi basis data untuk membantu mengidentifikasi setiap masalah yang mungkin mencegah tugas migrasi berjalan seperti yang diharapkan. Dengan menggunakan penilaian ini, Anda dapat mengidentifikasi berbagai potensi masalah sebelum Anda menjalankan tugas baru atau yang dimodifikasi. Untuk informasi lebih lanjut tentang bekerja dengan penilaian pra-migrasi, lihat [Mengaktifkan dan bekerja dengan penilaian perdana untuk tugas](#).

Mengubah skema

AWS DMS tidak melakukan konversi skema atau kode. Jika Anda ingin mengubah skema yang ada untuk mesin basis data yang berbeda, Anda dapat menggunakan AWS SCT. AWS SCT mengubah objek sumber, tabel, indeks, tampilan, pemicu, dan berbagai objek sistem lainnya ke dalam format bahasa definisi data target (DDL). Anda juga dapat menggunakan AWS SCT untuk mengubah sebagian besar kode aplikasi Anda, seperti PL/SQL atau TSQL, untuk bahasa target yang setara.

Anda bisa mendapatkan AWS SCT dengan mengunduh secara gratis dari AWS. Untuk informasi lebih lanjut tentang AWS SCT, lihat [Panduan Pengguna AWS SCT](#).

Jika titik akhir sumber dan target Anda berada di mesin database yang sama, Anda dapat menggunakan alat seperti Oracle SQL Developer, MySQL Workbench, atau 4 untuk memindahkan skema Anda. PgAdmin

Meninjau dokumentasi publik AWS DMS

Kami sangat menyarankan agar Anda pergi ke halaman dokumentasi publik AWS DMS untuk sumber dan titik akhir target sebelum migrasi pertama Anda. Dokumentasi ini dapat membantu Anda mengidentifikasi prasyarat untuk migrasi dan memahami keterbatasan saat ini sebelum memulai. Untuk informasi lebih lanjut, lihat [Bekerja dengan Titik akhir DMS AWS](#).

Selama migrasi, dokumentasi publik dapat membantu Anda memecahkan masalah apa pun dengan AWS DMS. Halaman pemecahan masalah dalam dokumentasi dapat membantu Anda menyelesaikan berbagai masalah umum menggunakan AWS DMS dan basis data titik akhir yang dipilih. Untuk informasi lebih lanjut, lihat [Pemecahan masalah tugas migrasi di AWS Database Migration Service](#).

Menjalankan bukti konsep

Untuk membantu menemukan masalah dengan lingkungan Anda pada fase awal migrasi basis data Anda, sebaiknya jalankan migrasi uji singkat. Melakukan hal ini juga dapat membantu Anda mengatur ketepatan waktu migrasi yang lebih realistis. Selain itu, Anda mungkin perlu menjalankan migrasi uji menskalakan penuh untuk mengukur apakah AWS DMS dapat menangani throughput basis data pada jaringan Anda. Selama waktu ini, kami merekomendasikan untuk menetapkan tolok ukur dan mengoptimalkan awal beban penuh Anda dan replikasi yang sedang berlangsung. Melakukan hal ini dapat membantu memahami latensi jaringan Anda dan mengukur performa secara keseluruhan.

Pada saat ini, Anda juga memiliki kesempatan untuk memahami profil data Anda dan seberapa basis data large Anda, termasuk berikut ini:

- Berapa banyak tabel yang berukuran Large, Medium, dan kecil.
- Bagaimana AWS DMS menangani tipe data dan konversi kumpulan karakter.
- Berapa banyak tabel yang memiliki kolom Large Object (LOB).
- Berapa lama waktu yang diperlukan untuk menjalankan migrasi pengujian.

Meningkatkan performa suatu migrasi AWS DMS

Sejumlah faktor memengaruhi performa Migrasi AWS DMS Anda:

- Ketersediaan sumber daya pada sumber.
- Throughput jaringan yang tersedia.
- Kapasitas sumber daya dari server replikasi.
- Kemampuan target untuk menyerap perubahan.
- Tipe dan distribusi data sumber.
- Jumlah objek yang akan dimigrasi.

Anda dapat meningkatkan performa dengan menggunakan beberapa atau semua praktik terbaik yang disebutkan berikut. Apakah Anda dapat menggunakan salah satu praktik ini tergantung pada kasus penggunaan spesifik Anda. Anda dapat menemukan beberapa keterbatasan berikut:

Penyediaan server replikasi yang tepat

AWS DMS adalah layanan terkelola yang berjalan di instans Amazon EC2. Layanan ini menghubungkan ke basis data sumber, membaca data sumber, melakukan format data untuk digunakan oleh basis data target, dan memuat data ke dalam basis data target.

Sebagian besar pemrosesan ini terjadi di memori. Namun, transaksi Large mungkin memerlukan beberapa buffering pada disk. Transaksi cache dan mencatat file juga ditulis pada disk. Pada beberapa bagian berikut, Anda dapat menemukan apa yang harus dipertimbangkan ketika memilih server replikasi Anda.

CPU

AWS DMS dirancang untuk migrasi heterogen, tetapi juga mendukung migrasi homogen. Untuk melakukan migrasi homogen, pertama mengubah setiap tipe sumber data untuk tipe data AWS DMS yang setara. Kemudian mengubah setiap jenis data AWS DMS ke jenis data target. Anda dapat menemukan beberapa referensi untuk konversi ini untuk setiap mesin basis data dalam Panduan Pengguna AWS DMS.

Untuk AWS DMS melakukan konversi ini secara optimal, CPU harus tersedia ketika konversi terjadi. Kelebihan beban CPU dan tidak memiliki sumber daya CPU yang cukup dapat mengakibatkan migrasi lambat, yang juga dapat menyebabkan efek samping lainnya.

Tipe instans replikasi

Beberapa kelas instans yang lebih kecil cukup untuk menguji layanan atau untuk migrasi kecil. Jika migrasi Anda melibatkan sejumlah tabel large, atau jika Anda berniat menjalankan beberapa tugas replikasi secara bersamaan, pertimbangkan untuk menggunakan salah satu instans yang lebih besar. Satu instans yang lebih besar dapat menjadi ide yang bagus karena layanan ini menghabiskan cukup banyak memori dan CPU.

Instans jenis T2 dirancang untuk memberikan performa dasar sedang dan kemampuan untuk meningkatkan performa yang secara signifikan lebih tinggi sesuai dengan yang dibutuhkan beban kerja Anda. Tujuannya adalah untuk beban kerja yang tidak menggunakan CPU penuh secara sering atau konsisten, tetapi kadang-kadang perlu secara berturut-turut. Instans T2 sangat cocok untuk beban kerja tujuan umum, seperti server web, lingkungan developer, dan basis data kecil. Jika Anda memecahkan masalah migrasi lambat dan menggunakan tipe instans T2, Anda dapat memeriksa metrik host Utilisasi CPU. Ini dapat menunjukkan kepada Anda jika Anda melampaui patokan dasar untuk Tipe instans itu.

Kelas instans C4 dirancang untuk memberikan tingkat performa prosesor tertinggi untuk beban kerja komputer intensif. Beban kerja tersebut mencapai performa paket per detik (PPS) yang jauh lebih tinggi, gangguan jaringan yang lebih rendah, dan latensi jaringan yang lebih rendah. AWS DMS dapat berupa CPU-intensif, terutama ketika melakukan migrasi heterogen dan replikasi seperti migrasi dari Oracle ke PostgreSQL. Instans C4 dapat menjadi pilihan yang bagus untuk situasi ini.

Kelas instans R4 adalah memori yang dioptimalkan untuk beban kerja memori-intensif. Migrasi yang sedang berlangsung atau replikasi dari sistem transaksi throughput tinggi menggunakan AWS DMS kadang dapat menghabiskan jumlah CPU dan memori large. Instans R4 mencakup lebih banyak memori per vCPU.

Support AWS DMS untuk kelas instans R5 dan C5

Kelas instans R5 adalah instans memori yang dioptimalkan yang dirancang untuk memberikan performa cepat untuk beban kerja yang memproses pengaturan data large dalam memori. Migrasi yang sedang berlangsung atau replikasi dari sistem transaksi throughput tinggi yang menggunakan AWS DMS kadang dapat menghabiskan jumlah CPU dan memori large. Instans R5 memberikan 5 persen memori tambahan per vCPU daripada R4 dan ukuran terbesar menyediakan memori 768 GiB. Selain itu, instans R5 memberikan harga 10 persen per peningkatan GiB dan ~20% peningkatan performa CPU pada R4.

Kelas instans C5 dioptimalkan untuk beban kerja intensif komputasi dan memberikan kinerja tinggi yang hemat biaya dengan harga rendah per rasio komputasi. Instans tersebut mencapai performa jaringan yang lebih tinggi. Adaptor Jaringan Elastis (ENA) menyediakan instans C5 dengan bandwidth jaringan hingga 25 Gbps dan hingga bandwidth khusus hingga 14 Gbps untuk Amazon EBS. AWS DMS dapat berupa CPU-intensif, terutama dalam performa migrasi dan replikasi heterogen seperti migrasi dari Oracle ke PostgreSQL. Instans C5 dapat menjadi pilihan yang bagus untuk situasi ini.

Penyimpanan

Tergantung pada kelas instans, server replikasi Anda dilengkapi dengan 50 GB ataupun 100 GB penyimpanan data. Penyimpanan ini digunakan untuk mencatat file dan setiap perubahan cache yang dikumpulkan selama beban itu. Jika sistem sumber sibuk atau melakukan transaksi large, Anda mungkin perlu meningkatkan penyimpanan. Jika Anda menjalankan beberapa tugas di server replikasi, Anda mungkin juga memerlukan peningkatan penyimpanan. Namun, jumlah default biasanya cukup.

Semua volume penyimpanan di AWS DMS adalah GP2 atau tujuan umum solid-state drives (SSD). Volume GP2 dilengkapi dengan performa dasar dari tiga operasi I/O per detik (IOPS), dengan kemampuan melonjak hingga 3.000 IOPS secara berangsur-angsur. Sebagai aturan praktis, sebaiknya memeriksa metrik `ReadIOPS` dan `WriteIOPS` untuk instans replikasi. Pastikan bahwa jumlah nilai ini tidak melampaui performa dasar untuk volume tersebut.

Multi-AZ

Memilih instans multi-AZ dapat melindungi migrasi Anda dari kegagalan penyimpanan. Sebagian besar migrasi bersifat sementara dan tidak dimaksudkan untuk dijalankan untuk jangka waktu yang lama. Jika Anda menggunakan AWS DMS untuk tujuan replikasi yang sedang berlangsung, memilih instans multi-AZ dapat meningkatkan ketersediaan Anda jika terjadi masalah penyimpanan.

Saat menggunakan instans replikasi AZ atau Multi-AZ tunggal selama FULL LOAD dan failover atau penggantian host terjadi, tugas pemuatan penuh diharapkan gagal. Anda dapat memulai ulang tugas dari titik kegagalan untuk tabel yang tersisa yang tidak selesai, atau berada dalam keadaan kesalahan.

Memuat beberapa tabel secara paralel

Secara default, AWS DMS memuat delapan tabel pada suatu waktu. Anda mungkin melihat beberapa peningkatan performa dengan meningkatkan hal ini sedikit ketika menggunakan server replikasi large, seperti `dms.c4.xlarge` atau instans yang lebih besar. Namun, pada titik tertentu, meningkatkan paralelisme ini akan mengurangi performa. Jika server replikasi Anda relatif kecil, seperti `dms.t2.medium`, kami menyarankan agar Anda mengurangi jumlah tabel yang dimuat secara paralel.

Untuk mengubah nomor ini di AWS Management Console, bukalah konsol tersebut, pilih Tugas, memilih untuk membuat atau mengubah tugas, dan kemudian memilih Pengaturan lanjutan. Dalam Pengaturan Penyetelan, mengubah pilihan Jumlah maksimum tabel pada beban dalam paralel.

Untuk mengubah nomor ini menggunakan AWS CLI, ubahlah parameter `MaxFullLoadSubTasks` dalam `TaskSettings`.

Menggunakan beban penuh paralel

Anda dapat menggunakan beban paralel dari sumber Oracle, Microsoft SQL Server, MySQL, Sybase, dan IBM Db2 LUW berdasarkan partisi dan subpartisi. Melakukan hal ini dapat meningkatkan durasi beban penuh secara keseluruhan. Selain itu, saat menjalankan tugas

migrasi AWS DMS, Anda dapat mempercepat migrasi tabel large atau yang dipartisi. Untuk melakukan ini, bagilah tabel ke dalam beberapa segmen dan muat beban segmen tersebut secara paralel dalam tugas migrasi yang sama.

Untuk menggunakan beban paralel, buatlah aturan pemetaan tabel tipe `table-settings` dengan pilihan `parallel-load`. Dalam aturan `table-settings`, tentukan kriteria pilihan untuk satu tabel atau beberapa tabel dengan beban yang ingin Anda muat secara paralel. Untuk menentukan kriteria pilihan, perlu pengaturan elemen `type` untuk `parallel-load` pada salah satu pengaturan berikut:

- `partitions-auto`
- `subpartitions-auto`
- `partitions-list`
- `ranges`
- `none`

Untuk informasi selengkapnya tentang pengaturan, lihat [Tabel dan koleksi pengaturan aturan dan operasi](#).

Bekerja dengan indeks, pemicu, dan kendala integritas referensial

Kendala indeks, pemicu, dan integritas referensial dapat memengaruhi performa migrasi Anda dan menyebabkan migrasi gagal. Bagaimana hal ini memengaruhi migrasi tergantung pada apakah tugas replikasi Anda merupakan beban tugas penuh atau tugas replikasi yang sedang berlangsung (tangkapan data perubahan, atau CDC).

Untuk tugas beban penuh, kami menyarankan agar Anda mengabaikan indeks kunci primer, indeks sekunder, kendala integritas referensial, dan pemicu bahasa manipulasi data (DLL). Atau Anda dapat menunda penyusunan semua itu sampai setelah tugas beban penuh selesai. Anda tidak memerlukan indeks selama tugas beban penuh, dan indeks dikenakan biaya pemeliharaan jika ada. Karena tugas beban penuh memuat beberapa kelompok tabel pada satu waktu, kendala integritas referensial dilanggar. Demikian pula, menyisipkan, memperbarui, dan menghapus pemicu dapat menyebabkan kesalahan, misalnya jika menyisipkan satu baris terpicu untuk tabel yang sebelumnya dimuat dalam jumlah banyak. Jenis pemicu lainnya juga memengaruhi performa karena pemrosesan tambahan.

Jika volume data relatif kecil dan waktu migrasi tambahan tidak menjadi pertimbangan, Anda dapat membangun indeks kunci primer dan sekunder sebelum tugas beban penuh. Selalu matikan batasan integritas referensial dan pemicu.

Untuk beban penuh ditambah tugas CDC, kami menyarankan agar Anda menambahkan indeks sekunder sebelum fase CDC. Karena AWS DMS menggunakan replikasi logis, pastikan bahwa indeks sekunder dalam support operasi DLL berada pada tempatnya untuk mencegah pemindaian tabel penuh. Anda dapat menjeda tugas replikasi sebelum fase CDC untuk membangun indeks dan membuat batasan integritas referensial sebelum memulai ulang tugas.

Anda harus mengaktifkan pemicu tepat sebelum cutover.

Matikan cadangan dan pendataan log transaksi

Saat bermigrasi ke basis data Amazon RDS, sebaiknya matikan cadangan dan Multi-AZ pada target sampai Anda siap untuk melakukan cutover. Demikian pula, ketika bermigrasi ke sistem selain Amazon RDS, mematikan setiap pendataan log pada target sampai setelah cutover biasanya merupakan ide bagus.

Menggunakan beberapa tugas

Kadang-kadang menggunakan beberapa tugas untuk satu migrasi dapat meningkatkan performa. Jika Anda memiliki kumpulan tabel yang tidak turut serta dalam transaksi umum, Anda mungkin dapat membagi migrasi menjadi beberapa tugas. Konsistensi transaksional dipertahankan dalam suatu tugas, jadi penting bila beberapa tabel dalam tugas terpisah tidak turut serta dalam transaksi umum. Selain itu, setiap tugas secara bebas membaca pengaliran transaksi, jadi berhati-hatilah agar tidak memberikan terlalu banyak tekanan pada basis data sumber.

Anda dapat menggunakan beberapa tugas untuk membuat streaming replikasi terpisah. Dengan melakukan ini, Anda dapat memparalelkan bacaan pada sumber, proses pada instans replikasi, dan penulisan pada basis data target.

Mengoptimalkan pemrosesan perubahan

Secara default, proses AWS DMS berubah dalam mode transaksional, yang mempertahankan integritas transaksional. Jika Anda mampu penyimpangan sementara dalam integritas transaksional, Anda dapat menggunakan batch dioptimalkan menerapkan pilihan sebagai gantinya. Opsi ini secara efisien mengelompokkan transaksi dan menerapkannya dalam batch untuk tujuan efisiensi. Menggunakan batch dioptimalkan menerapkan opsi hampir selalu melanggar referensial integritas kendala. Jadi kami menyarankan agar Anda menonaktifkan kendala ini selama proses migrasi dan mengaktifkannya lagi sebagai bagian dari proses cutover.

Menggunakan server nama on-premise Anda sendiri

Biasanya, instans replikasi AWS DMS menggunakan resolver Sistem Nama Domain (DNS) dalam instans Amazon EC2 untuk menyelesaikan titik akhir domain. Namun, Anda dapat menggunakan server nama on-premise Anda sendiri untuk menyelesaikan titik-titik akhir tertentu jika Anda menggunakan Amazon Route 53 Resolver. Dengan perangkat ini, Anda dapat melakukan kueri antara on-premise dan AWS menggunakan titik akhir masuk (inbound) dan keluar (outbound), aturan penerusan, dan koneksi privat. Manfaat menggunakan server nama on-premise mencakup peningkatan keamanan dan kemudahan penggunaan di balik firewall.

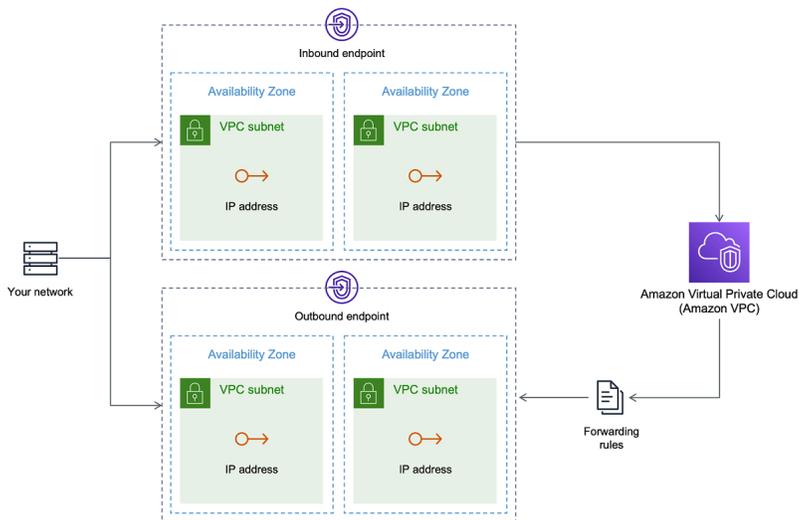
Jika Anda memiliki titik akhir masuk (inbound), Anda dapat menggunakan kueri DNS yang berasal dari on-premise untuk menyelesaikan domain host AWS. Untuk mengkonfigurasi titik akhir, tetapkan alamat IP di setiap subnet yang ingin Anda berikan resolver. Untuk membangun konektivitas antara infrastruktur DNS on-premise dan AWS, gunakan AWS Direct Connect atau jaringan pribadi virtual (VPN).

Connect titik akhir keluar ke server nama on-premise Anda. Server nama hanya memberi kemudahan untuk mengakses ke alamat IP yang disertakan dalam daftar yang mengizinkan maksud tersebut dan diatur dalam titik akhir keluar. Alamat IP server nama Anda adalah alamat IP target. Ketika Anda memilih grup keamanan untuk titik akhir keluar, pilihlah grup keamanan sama yang digunakan oleh instans replikasi.

Untuk meneruskan domain pilihan ke server nama, gunakan aturan penerusan. Titik akhir keluar dapat menangani beberapa aturan penerusan. Lingkup aturan penerusan adalah virtual private cloud (VPC) Anda. Dengan menggunakan aturan penerusan yang terkait dengan VPC, Anda dapat memiliki persediaan bagian Cloud AWS yang terisolasi secara logis. Dari bagian yang terisolasi secara logis ini, Anda dapat meluncurkan sumber daya AWS dalam jaringan virtual.

Anda dapat melakukan konfigurasi domain yang dihosting dalam infrastruktur DNS on-premise sebagai aturan penerusan bersyarat yang mengatur kueri DNS keluar. Ketika suatu kueri dibuat untuk salah satu domain tersebut, beberapa aturan menjadikan pemicu sebagai upaya untuk meneruskan permintaan DNS ke server yang dikonfigurasi dengan aturan tersebut. Sekali lagi, diperlukan koneksi privat melalui AWS Direct Connect atau VPN.

Diagram berikut menunjukkan arsitektur Route 53 Resolver.



Untuk informasi selengkapnya tentang penggunaan Route 53 DNS Resolver, lihat [Memulai dengan Route 53 Resolver](#) dalam Panduan Developer Amazon Route 53.

Menggunakan Amazon Route 53 Resolver dengan AWS DMS

Anda dapat membuat server nama on premise untuk AWS DMS untuk menyelesaikan titik akhir menggunakan [Amazon Route 53 Resolver](#).

Untuk membuat server nama on premise untuk AWS DMS berdasarkan Route 53

1. Masuk ke AWS Management Console dan bukalah konsol Route 53 di <https://console.aws.amazon.com/route53/>.
2. Pada konsol Route 53, sebaiknya memilih Wilayah AWS tempat Anda ingin melakukan konfigurasi Route 53 Resolver. Route 53 Resolver adalah khusus untuk suatu Wilayah.
3. Memilih arah kueri—masuk, keluar, atau keduanya.
4. Menyediakan konfigurasi kueri masuk:
 - a. Memasukkan nama titik akhir dan memilih VPC.
 - b. Menetapkan satu atau lebih subnet dari dalam VPC (misalnya, memilih dua untuk ketersediaan).
 - c. Menetapkan alamat IP tertentu untuk digunakan sebagai titik-titik akhir, atau memiliki Route 53 Resolver yang menentukannya secara otomatis.
5. Membuat aturan untuk domain on premise Anda sehingga beban kerja di dalam VPC dapat memberikan rute kueri DNS ke infrastruktur DNS Anda.
6. Memasukkan satu atau beberapa alamat IP untuk server DNS on-premise Anda.

7. Kirim aturan Anda.

Ketika semuanya dibuat, VPC Anda dikaitkan dengan aturan masuk dan keluar Anda dan lalu lintas perutean dapat mulai.

Untuk informasi selengkapnya tentang Route 53 Resolver, lihat [Memulai dengan Route 53 Resolver](#) dalam Panduan Developer Amazon Route 53.

Melakukan migrasi objek biner large (LOB)

Secara umum, AWS DMS melakukan migrasi data LOB dalam dua tahap:

1. AWS DMS menciptakan baris baru dalam tabel target dan mengisi baris tersebut dengan semua data kecuali nilai LOB terkait.
2. AWS DMS memperbarui baris dalam tabel target dengan LOB data.

Proses migrasi untuk LOB mensyaratkan bahwa, selama migrasi, semua kolom LOB pada tabel target harus tidak dapat dibatalkan. Hal ini sedemikian rupa sehingga bahkan jika kolom LOB bukan tidak dapat dibatalkan pada tabel sumber. Jika AWS DMS menciptakan tabel target, maka menetapkan kolom LOB supaya tidak dapat dibatalkan secara default. Dalam beberapa kasus, Anda mungkin membuat tabel target menggunakan beberapa mekanisme lain, seperti impor atau ekspor. Dalam kasus tersebut, pastikan bahwa kolom LOB tidak dapat dibatalkan sebelum Anda mulai tugas migrasi.

Persyaratan ini memiliki satu pengecualian. Misalkan Anda melakukan migrasi homogen dari sumber Oracle ke target Oracle, dan Anda memilih Mode Lob terbatas. Dalam hal ini, seluruh baris diisi sekaligus, termasuk nilai-nilai LOB. Untuk kasus seperti itu, AWS DMS dapat membuat kolom LOB tabel target dengan kendala bukan tidak dapat dibatalkan, jika diperlukan.

Menggunakan mode LOB terbatas

AWS DMS menggunakan dua metode yang menyeimbangkan performa dan kemudahan ketika migrasi Anda berisi nilai-nilai LOB:

1. Mode LOB terbatas melakukan migrasi semua nilai LOB hingga batas pengukuran yang ditentukan oleh pengguna (default adalah 32 KB). Nilai LOB yang lebih besar dari batas pengukuran harus dimigrasi secara manual. Mode LOB terbatas, default untuk semua tugas migrasi, biasanya

memberikan performa terbaik. Namun, pastikan bahwa pengaturan parameter Pengukuran LOB maksimal itu benar. Mengatur parameter ini pada pengukuran LOB terbesar untuk semua tabel Anda.

2. Mode LOB penuh melakukan migrasi semua data LOB dalam tabel Anda, terlepas dari ukurannya. Mode LOB penuh memberikan kemudahan memindahkan semua data LOB dalam tabel Anda, tetapi proses ini dapat memiliki dampak yang signifikan pada performa.

Untuk beberapa mesin basis data, seperti PostgreSQL, AWS DMS memperlakukan tipe data JSON seperti LOBs. Pastikan bahwa jika Anda memilih Mode LOB terbatas, pengukuran LOB maksimal diatur pada nilai yang tidak menyebabkan data JSON dipotong.

AWS DMS menyediakan support penuh untuk menggunakan tipe data objek large (BLOB, CLOB, dan NCLOB). Titik akhir sumber berikut memiliki support LOB penuh:

- Oracle
- Microsoft SQL Server
- ODBC

Titik akhir target berikut memiliki support LOB penuh:

- Oracle
- Microsoft SQL Server

Titik akhir target berikut memiliki support LOB terbatas. Anda tidak dapat menggunakan pengukuran LOB tak terbatas untuk titik akhir target ini.

- Amazon Redshift
- Amazon S3

Untuk titik akhir yang memiliki support LOB penuh, Anda juga dapat menetapkan batas pengukuran untuk LOB tipe data.

Peningkatan performa LOB

Sementara melakukan migrasi data LOB, Anda dapat menentukan pengaturan optimasi LOB yang berbeda berikut.

Pengaturan LOB per tabel

Menggunakan pengaturan LOB per tabel, Anda dapat mengganti pengaturan LOB tingkat tugas untuk beberapa atau semua tabel Anda. Untuk melakukannya, tentukan `lob-settings` di aturan `table-settings`. Berikut ini adalah contoh tabel yang mencakup beberapa nilai LOB large.

```
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 || '000000000000000000000000000000000000';
END LOOP;
INSERT INTO TEST_CLOB (ID, C1, C2) VALUES (0, bigtextstring,'AnyValue');
END;
/
SELECT * FROM TEST_CLOB;
COMMIT
```

Berikutnya, membuat tugas migrasi dan memodifikasi penanganan LOB untuk tabel Anda dengan menggunakan aturan baru `lob-settings`. Nilai `bulk-max-siz` menentukan pengukuran LOB maksimum (KB). Ini terpotong jika lebih besar dari pengukuran yang ditentukan.

```
{
  "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"
}
]
}
```

Bahkan jika tugas ini AWS DMS dibuat dengan `FullLobMode : true`, beberapa pengaturan LOB per tabel mengarahkan AWS DMS untuk memotong data LOB dalam tabel khusus ini pada 16.000. Anda dapat memeriksa catatan tugas untuk mengonfirmasi ini.

```
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
```

Pengaturan LOB sebaris

Saat Anda membuat suatu tugas AWS DMS, mode LOB menentukan bagaimana LOB ditangani.

Dengan mode LOB penuh dan mode LOB terbatas, masing-masing memiliki manfaat dan kekurangan tersendiri. Mode LOB sebaris menggabungkan keuntungan dari kedua mode LOB penuh dan mode LOB terbatas.

Anda dapat menggunakan mode LOB sebaris ketika Anda perlu mereplikasi baik LOB kecil maupun large, dan sebagian besar LOB kecil. Ketika Anda memilih opsi ini, selama beban penuh, tugas AWS DMS mentransfer baris LOB kecil, yang lebih efisien. Tugas AWS DMS mentransfer LOB large dengan performa pencarian dari tabel sumber.

Selama pemrosesan perubahan, LOB kecil dan large direplikasi dengan performa pencarian dari tabel sumber.

Bila Anda menggunakan modus LOB sebaris, tugas AWS DMS memeriksa semua pengukuran LOB untuk menentukan mana yang ditransfer sebaris. LOB yang lebih besar dari pengukuran yang ditentukan direplikasi dengan menggunakan mode LOB penuh. Oleh karena itu, jika Anda

tahu bahwa sebagian besar LOB lebih besar dari pengaturan yang ditentukan, lebih baik tidak menggunakan opsi ini. Sebaliknya, mengizinkan pengukuran LOB yang tak terbatas.

Anda mengkonfigurasi opsi ini menggunakan atribut dalam pengaturan tugas, `InlineLobMaxSize`, yang hanya tersedia saat `FullLobMode` dilakukan pengaturan ke `true`. Nilai default untuk `InlineLobMaxSize` adalah 0 dan kisarannya adalah 1 —102400 kilobyte (100 MB).

Misalnya, Anda dapat menggunakan perintah berikut: pengaturan tugas AWS DMS. Di sini, pengaturan `InlineLobMaxSize` ke nilai 5 menghasilkan semua LOB yang lebih kecil dari atau sama dengan 5 KiB (5.120 byte) ditransfer sebaris.

```
{
  "TargetMetadata": {
    "TargetSchema": "",
    "SupportLobs": true,
    "FullLobMode": true,
    "LobChunkSize": 64,
    "LimitedSizeLobMode": false,
    "LobMaxSize": 32,
    "InlineLobMaxSize": 5,
    "LoadMaxFileSize": 0,
    "ParallelLoadThreads": 0,
    "ParallelLoadBufferSize": 0,
    "BatchApplyEnabled": false,
    "TaskRecoveryTableEnabled": false},
  . . .
}
```

Meningkatkan performa ketika melakukan migrasi tabel-tabel large menggunakan filter baris

Untuk meningkatkan kinerja saat memigrasikan tabel besar, pisahkan migrasi menjadi lebih dari satu tugas. Untuk membagi migrasi menjadi beberapa tugas menggunakan filter baris, gunakan satu kunci atau kunci partisi. Misalnya, jika Anda memiliki ID kunci primer integer dari 1 hingga 8.000.000, Anda dapat membuat delapan tugas menggunakan filter baris untuk melakukan migrasi masing-masing 1 juta catatan.

Untuk menerapkan pemfilteran baris di konsol:

1. Buka AWS Management Console.

2. Pilih Tugas, dan buat tugas baru.
3. Memilih tab Pemetaan tabel, dan memperluas Aturan Pilihan.
4. Memilih Menambahkan aturan pilihan baru. Anda sekarang dapat menambahkan filter kolom dengan kurang dari atau sama dengan, lebih besar dari atau sama dengan, sama dengan, atau kondisi rentang antara dua nilai. Untuk informasi selengkapnya tentang pemfilteran kolom, lihat [Menentukan pemilihan tabel dan transformasi aturan dari konsol](#).

Jika Anda memiliki tabel partisi besar yang dipartisi berdasarkan tanggal, Anda dapat memigrasikan data berdasarkan tanggal. Misalnya, anggap bahwa Anda memiliki tabel yang dipartisi berdasarkan bulan, dan hanya data bulan saat ini yang diperbarui. Dalam hal ini, Anda dapat membuat tugas beban penuh untuk setiap partisi bulanan statis dan membuat beban penuh ditambah tugas CDC untuk partisi yang sedang diperbarui.

Jika tabel Anda memiliki kunci primer satu kolom atau indeks unik, Anda dapat membuat AWS DMS tugas Anda mengelompokkan tabel menggunakan beban paralel dari jenis rentang untuk memuat data secara paralel. Untuk informasi selengkapnya, lihat [Tabel dan koleksi pengaturan aturan dan operasi](#).

Replikasi yang sedang berlangsung

AWS DMS menyediakan replikasi data yang sedang berlangsung, menjaga sumber dan target basis data tetap sinkron. Ini mereplikasi hanya jumlah terbatas bahasa definisi data (DDL). AWS DMS tidak mengembangkan item seperti indeks, pengguna, hak istimewa, prosedur yang disimpan, dan perubahan basis data lainnya yang tidak terkait langsung dengan data tabel.

Jika Anda berencana untuk menggunakan replikasi yang sedang berlangsung, aturlah opsi Multi-AZ saat membuat instans replikasi Anda. Dengan memilih Multi-AZ, Anda mendapatkan ketersediaan yang tinggi dan failover support untuk instans replikasi. Namun, opsi ini dapat berdampak pada performa dan dapat memperlambat replikasi sementara menerapkan perubahan sistem target.

Sebelum Anda memutakhirkan sumber atau target basis data, kami menyarankan agar Anda menghentikan setiap tugas AWS DMS yang berjalan di basis data ini. Lanjutkan tugas setelah pemutakhiran sistem selesai.

Selama replikasi yang sedang berlangsung, sangat penting untuk mengidentifikasi bandwidth jaringan antara sistem basis data sumber dan instans replikasi AWS DMS. Pastikan bahwa jaringan tidak menyebabkan kemacetan selama replikasi berlangsung.

Penting juga untuk mengidentifikasi tingkat perubahan dan hasil dari mencatat arsip per jam pada sistem basis data sumber Anda. Melakukan hal ini dapat membantu Anda untuk memahami throughput yang mungkin Anda dapatkan selama replikasi yang sedang berlangsung.

Mengurangi beban pada basis data sumber Anda

AWS DMS menggunakan beberapa sumber daya pada basis data sumber Anda. Selama tugas beban penuh, AWS DMS melakukan pemindaian tabel lengkap dari tabel sumber untuk setiap tabel yang diproses secara paralel. Dan juga, setiap tugas yang Anda buat sebagai bagian dari migrasi kueri sumber untuk perubahan sebagai bagian dari proses CDC. Supaya AWS DMS melakukan CDC untuk beberapa sumber, seperti Oracle, Anda mungkin perlu meningkatkan jumlah data yang ditulis dengan mencatat perubahan basis data Anda.

Jika mengetahui bahwa Anda membebani basis data sumber Anda, kurangi jumlah tugas atau tabel untuk setiap tugas migrasi Anda. Setiap tugas mendapat perubahan sumber secara terpisah, sehingga menggabungkan beberapa tugas dapat mengurangi perubahan yang menangkap beban kerja.

Mengurangi kemacetan pada basis data target Anda

Selama migrasi, cobalah menghapus proses yang bersaing untuk sumber daya tulis pada basis data target Anda:

- Matikan pemicu yang tidak perlu.
- Matikan indeks sekunder selama beban awal dan mengubahnya kembali kemudian selama replikasi yang sedang berlangsung.
- Dengan basis data Amazon RDS, ide yang baik bila mematikan backup dan multi-AZ sampai cutover.
- Ketika melakukan migrasi ke sistem non-RDS, ide yang baik bila menonaktifkan setiap catatan pada target sampai cutover.

Menggunakan validasi data selama migrasi

Untuk memastikan bahwa data Anda dimigrasi secara akurat dari sumber ke target, kami sangat menyarankan agar Anda menggunakan validasi data. Jika Anda mengaktifkan validasi data untuk suatu tugas, AWS DMS mulai membandingkan sumber dan target data segera setelah beban penuh dilakukan untuk tabel.

Validasi data bekerja dengan basis data berikut di mana pun AWS DMS mendukungnya sebagai sumber dan target titik akhir:

- Oracle
- PostgreSQL
- MySQL
- MariaDB
- Microsoft SQL Server
- Amazon Aurora Edisi Kompatibel MySQL
- Amazon Aurora Edisi Kompatibel PostgreSQL
- IBM Db2 LUW
- Amazon Redshift

Untuk informasi selengkapnya, lihat [Validasi data AWS DMS](#).

Pemantauan tugas AWS DMS menggunakan metrik

Anda memiliki beberapa opsi pemantauan metrik untuk tugas Anda dengan menggunakan konsol AWS DMS tersebut:

Metrik host

Anda dapat menemukan metrik host di tab CloudWatch metrik untuk setiap instance replikasi tertentu. Di sini, Anda dapat memantau apakah instans replikasi Anda diukur secara tepat.

Metrik tugas replikasi

Metrik untuk tugas replikasi, termasuk perubahan masuk dan komit, dan latensi antara host replikasi dan basis data sumber/target dapat ditemukan di tab metrik untuk setiap tugas tertentu. CloudWatch

Metrik tabel

Anda dapat menemukan metrik tabel individual pada tab Statistik tabel untuk setiap tugas individu. Metrik ini mencakup angka-angka berikut:

- Baris-baris dimuat selama beban penuh.
- Menyisipkan, memperbarui, dan menghapus sejak tugas dimulai.
- Operasi DDL sejak tugas dimulai.

Untuk informasi selengkapnya tentang pemantauan metrik, lihat [Pemantauan Tugas AWS DMS](#).

Tindakan dan notifikasi

AWS DMS menggunakan Amazon SNS untuk memberikan notifikasi ketika tindakan AWS DMS terjadi, misalnya penciptaan atau penghapusan instans replikasi. Anda dapat bekerja dengan notifikasi ini dalam bentuk apa pun yang didukung oleh Amazon SNS untuk Wilayah AWS. Ini dapat mencakup pesan email, pesan teks, atau panggilan pada titik akhir HTTP.

Untuk informasi lebih lanjut, lihat [Bekerja dengan acara Amazon SNS dan pemberitahuan diAWS Database Migration Service](#).

Menggunakan cara dengan mencatat tugas untuk memecahkan masalah migrasi

Dalam beberapa kasus, AWS DMS dapat mengalami berbagai masalah yang memunculkan peringatan atau pesan kesalahan hanya dengan mencatat tugas. Secara khusus, masalah pemotongan data atau penolakan baris karena pelanggaran kunci asing hanya ditulis dengan mencatat tugas. Oleh karena itu, pastikan untuk meninjau setelah mencatat tugas ketika melakukan migrasi basis data. Untuk melihat log tugas, konfigurasi Amazon CloudWatch sebagai bagian dari pembuatan tugas.

Untuk informasi selengkapnya, lihat [Memantau tugas replikasi menggunakan Amazon CloudWatch](#).

Memecahkan masalah tugas replikasi dengan Time Travel

Untuk memecahkan masalah AWS DMS migrasi, Anda dapat bekerja dengan Time Travel. Untuk informasi lebih lanjut tentang Perjalanan Waktu, lihat [Pengaturan tugas Perjalanan Waktu](#).

Saat Anda bekerja dengan Time Travel, perhatikan pertimbangan berikut:

- Untuk menghindari overhead pada instance replikasi DMS, aktifkan Time Travel hanya untuk tugas yang memerlukan debugging.

- Saat Anda menggunakan Time Travel untuk memecahkan masalah tugas replikasi yang mungkin berjalan selama beberapa hari, pantau metrik instance replikasi untuk overhead sumber daya. Pendekatan ini berlaku terutama dalam kasus di mana beban transaksi tinggi berjalan pada database sumber untuk jangka waktu yang lama. Untuk detail selengkapnya, lihat [Pemantauan Tugas AWS DMS](#).
- Saat pengaturan tugas Perjalanan Waktu `EnableRawData` disetel ke `true`, penggunaan memori tugas selama replikasi DMS mungkin lebih tinggi daripada saat Perjalanan Waktu tidak diaktifkan. Jika Anda mengaktifkan Perjalanan Waktu untuk waktu yang lama, pantau tugas Anda.
- Saat ini, Anda dapat mengaktifkan Perjalanan Waktu hanya di tingkat tugas. Perubahan pada semua tabel dicatat di log Perjalanan Waktu. Jika Anda memecahkan masalah untuk tabel tertentu dalam database dengan volume transaksi tinggi, buat tugas terpisah.

Mengubah pengguna dan skema untuk target Oracle

Bila Anda adalah pengguna Oracle sebagai target, AWS DMS melakukan migrasi data ke skema yang dimiliki oleh pengguna titik akhir target.

Misalnya, anggaplah Anda melakukan migrasi skema bernama PERFDATA pada titik akhir target Oracle, dan bahwa nama pengguna titik akhir target adalah MASTER. AWS DMS menghubungkan ke target Oracle sebagai MASTER dan mengisi skema MASTER dengan objek basis data dari PERFDATA.

Untuk membatalkan perilaku ini, berikan perubahan skema. Misalnya, untuk melakukan migrasi objek skema PERFDATA ke skema PERFDATA pada titik akhir target, gunakan perubahan berikut.

```
{
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "object-locator": {
    "schema-name": "PERFDATA"
  },
  "rule-target": "schema",
  "rule-action": "rename",
  "value": "PERFDATA"
}
```

Untuk informasi lebih lanjut tentang perubahan, lihat [Menentukan pemilihan tabel dan transformasi aturan menggunakan JSON](#).

Mengubah tempat penyimpanan tabel dan indeks untuk target Oracle

Bila menggunakan Oracle sebagai target, AWS DMS melakukan migrasi semua tabel dan indeks ke ruang penyimpanan default dalam target. Misalnya, anggap bahwa sumber Anda adalah mesin basis data selain Oracle. Semua tabel dan indeks target dimigrasi ke tempat penyimpanan default yang sama.

Untuk membatalkan perilaku ini, berikan perubahan tempat penyimpanan yang sesuai. Sebagai contoh, misalkan Anda ingin melakukan migrasi tabel dan indeks ke tempat penyimpanan tabel dan indeks di target Oracle yang diberi nama sesuai skema dalam sumber. Dalam hal ini, Anda dapat menggunakan perubahan serupa dengan yang berikut. Di sini, skema dalam sumber diberi nama INVENTORY dan tempat penyimpanan tabel dan indeks yang sesuai dalam target diberi nama INVENTORYTBL dan 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"
}
```

```
}
```

Untuk informasi lebih lanjut tentang perubahan, lihat [Menentukan pemilihan tabel dan transformasi aturan menggunakan JSON](#).

Apabila Oracle adalah sumber dan target, Anda dapat mempertahankan tugas tempat penyimpanan tabel atau indeks yang ada dengan menetapkan atribut koneksi tambahan sumber Oracle `enableHomogenousTablespace=true`. Untuk informasi lebih lanjut, lihat [Pengaturan titik akhir saat menggunakan Oracle sebagai sumber AWS DMS](#).

Pemutakhiran versi instans replikasi

AWS secara berkala merilis versi baru dari perangkat lunak mesin replikasi AWS DMS, dengan fitur baru dan peningkatan performa. Setiap versi perangkat lunak mesin replikasi memiliki nomor versi sendiri. Sangat penting untuk menguji versi AWS DMS instans replikasi Anda yang menjalankan beban kerja produksi sebelum Anda memutakhirkan instans replikasi Anda ke versi yang lebih baru. Untuk informasi lebih lanjut tentang pemutakhiran versi, lihat [AWS Catatan rilis DMS](#).

Memahami biaya migrasi Anda

AWS Database Migration Service membantu Anda melakukan migrasi basis data ke AWS secara mudah dan aman dengan biaya rendah. Anda hanya membayar untuk instans replikasi Anda dan penyimpanan setelah mencatat informasi tambahan. Setiap instans migrasi basis data termasuk penyimpanan yang cukup untuk ruang tukar, catatan replikasi, dan cache data untuk sebagian besar replikasi dan transfer data masuk adalah gratis.

Anda mungkin memerlukan lebih banyak sumber daya selama beban awal atau selama waktu unggah puncak. Anda dapat memantau penggunaan sumber daya instans replikasi menggunakan metrik cloud watch. Anda kemudian dapat menaikkan skala dan menurunkan skala ukuran instans replikasi berdasarkan penggunaan.

Untuk informasi lebih lanjut tentang memperkirakan biaya migrasi Anda, lihat:

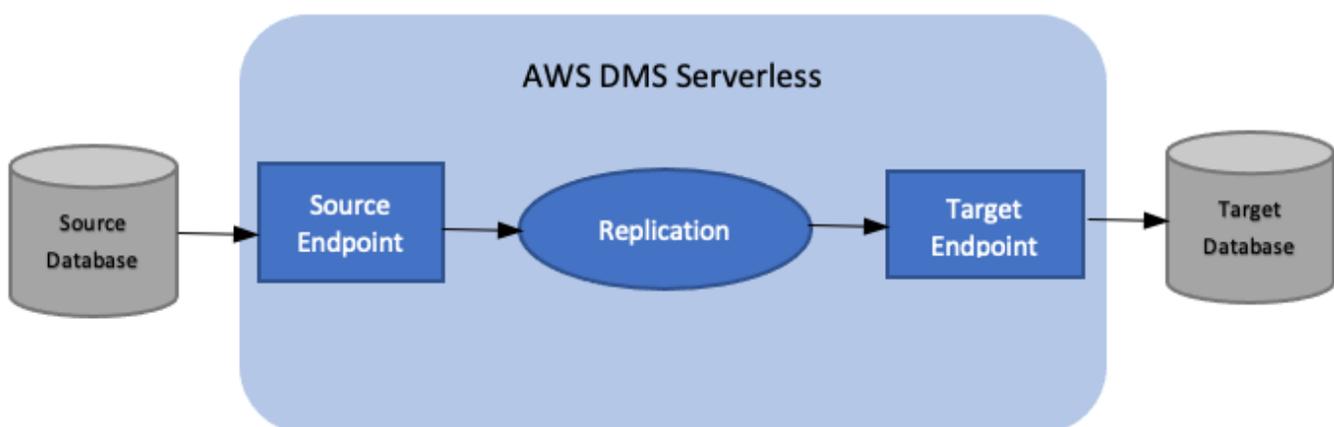
- [AWS Database Migration Service harga](#)
- [AWS Kalkulator Harga](#)

Bekerja dengan Tanpa AWS DMS Server

AWS DMS Tanpa server adalah fitur yang menyediakan penyediaan otomatis, penskalaan, ketersediaan tinggi bawaan, dan model pay-for-use penagihan, untuk meningkatkan kelincuhan operasi dan mengoptimalkan biaya Anda. Fitur Tanpa Server menghilangkan tugas manajemen instans replikasi seperti estimasi kapasitas, penyediaan, pengoptimalan biaya, dan mengelola versi mesin replikasi dan patching.

Dengan AWS DMS Tanpa Server, mirip dengan fungsionalitas saat ini AWS DMS (disebut dalam dokumen ini sebagai AWS DMS Standar), Anda membuat koneksi sumber dan target menggunakan titik akhir. Setelah membuat titik akhir sumber dan target, Anda membuat konfigurasi replikasi, yang mencakup pengaturan konfigurasi untuk replikasi yang diberikan. Anda dapat mengelola replikasi dengan memulai, menghentikan, memodifikasi, atau menghapusnya. Setiap replikasi memiliki pengaturan yang dapat Anda konfigurasi sesuai dengan persyaratan migrasi database Anda. Anda menentukan pengaturan ini menggunakan file JSON atau AWS DMS bagian dari file. AWS Management Console Untuk informasi selengkapnya tentang pengaturan replikasi, lihat [Bekerja dengan titik AWS DMS akhir](#). Setelah memulai replikasi, AWS DMS tanpa server terhubung ke database sumber dan mengumpulkan metadata database untuk menganalisis beban kerja replikasi. Dengan menggunakan metadata ini, AWS DMS menghitung dan menyediakan kapasitas yang diperlukan dan memulai replikasi data.

Diagram berikut menunjukkan proses replikasi AWS DMS Tanpa Server.



Note

AWS DMS Tanpa server menggunakan versi mesin default. Untuk informasi tentang versi mesin default, lihat [Catatan rilis](#).

Lihat topik berikut untuk menemukan detail selengkapnya tentang Tanpa AWS DMS Server.

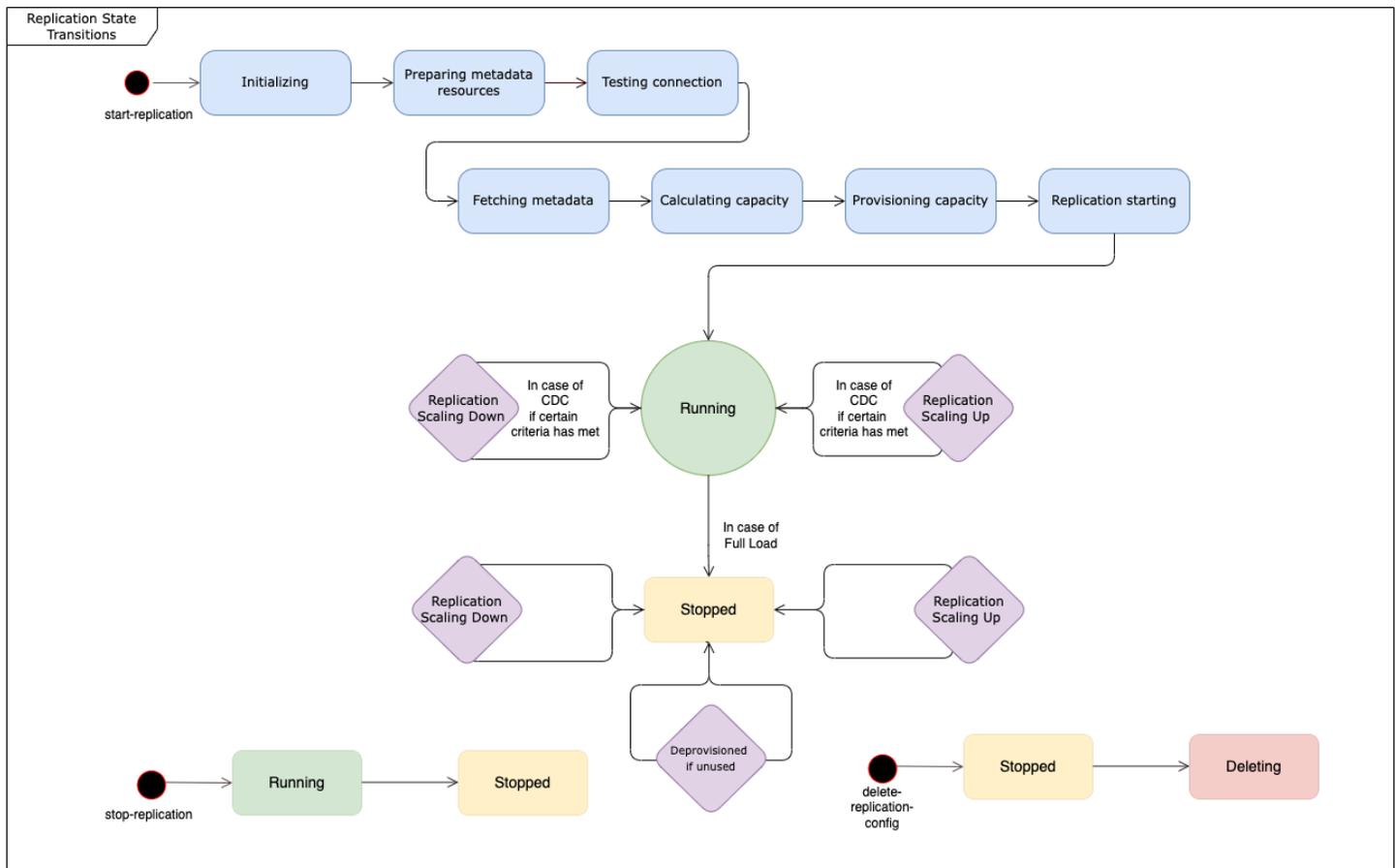
Topik

- [AWS DMS Komponen tanpa server](#)
- [AWS DMS Batasan tanpa server](#)

AWS DMS Komponen tanpa server

Untuk mengelola sumber daya yang diperlukan untuk melakukan replikasi, AWS DMS Serverless memiliki status granular yang mengungkapkan berbagai tindakan internal yang diambil oleh layanan. Saat Anda memulai replikasi, AWS DMS Tanpa Server menghitung beban kapasitas, menyediakan kapasitas yang dihitung, dan memulai replikasi data sesuai dengan status replikasi berikut.

Diagram berikut menunjukkan transisi status untuk replikasi Tanpa AWS DMS Server.



- Keadaan pertama setelah Anda memulai replikasi adalah Inisialisasi. Dalam keadaan ini, semua parameter yang diperlukan diinisialisasi.
- Negara-negara yang segera mengikuti termasuk Mempersiapkan Sumber Daya Metadata, Menguji Koneksi, dan Mengambil Metadata. Di negara bagian ini, AWS DMS Tanpa Server terhubung ke basis data sumber Anda untuk mendapatkan informasi yang diperlukan untuk memprediksi kapasitas yang dibutuhkan.
 - Saat status replikasi adalah Testing Connection, AWS DMS Serverless memverifikasi bahwa koneksi ke basis data sumber dan target Anda berhasil disiapkan.
 - Status replikasi setelah Testing Connection adalah Fetching Metadata. Di sini, AWS DMS mengambil informasi yang diperlukan untuk menghitung kapasitas.
 - Setelah AWS DMS mengambil informasi yang diperlukan, keadaan berikutnya adalah Menghitung Kapasitas. Di sini, sistem menghitung ukuran sumber daya yang mendasari yang diperlukan untuk melakukan replikasi.
- Transisi status setelah Menghitung Kapasitas adalah Kapasitas Penyediaan. Sementara replikasi dalam keadaan ini, AWS DMS Tanpa Server menginisialisasi sumber daya komputasi yang mendasarinya.

- Status replikasi setelah semua sumber daya berhasil disediakan adalah Replikasi Mulai. Dalam keadaan ini, AWS DMS Tanpa Server memulai replikasi data. Fase replikasi meliputi:
 - Beban penuh: Pada fase ini, DMS mereplikasi penyimpanan data sumber seperti ketika replikasi dimulai.
 - CDC (initial): Pada fase ini, DMS mereplikasi perubahan pada penyimpanan data sumber yang terjadi selama fase Full Load. DMS hanya menjalankan fase ini jika pengaturan `StopTaskCachedChangesNotApplied` tugasnya. `false`
 - CDC (sedang berlangsung): Setelah fase CDC awal, DMS mereplikasi perubahan pada database sumber saat terjadi. DMS hanya terus menjalankan replikasi setelah fase CDC awal jika pengaturan `StopTaskCachedChangesApplied` tugasnya. `false`
- Keadaan terakhir adalah Running. Dalam status Running, replikasi data sedang berlangsung.
- Replikasi yang Anda hentikan memasuki status Berhenti. Anda dapat memulai ulang replikasi yang dihentikan dalam keadaan berikut:
 - Anda tidak dapat memulai ulang replikasi yang DMS telah deprovisioned.
 - Anda dapat memulai ulang replikasi CDC saja atau beban penuh dan CDC yang dihentikan menggunakan tindakan. [StartReplication](#) Anda tidak dapat memulai ulang replikasi yang dihentikan menggunakan konsol.
 - Anda tidak dapat memulai ulang replikasi berhenti yang menggunakan PostgreSQL sebagai mesin.

Topik ini berisi bagian-bagian berikut.

- [Versi Mesin yang Didukung](#)
- [Membuat replikasi tanpa server](#)
- [Memodifikasi replikasi AWS DMS tanpa server](#)
- [Konfigurasi Komputasi](#)
- [Memahami penskalaan otomatis tanpa server AWS DMS](#)
- [Memantau AWS DMS replikasi tanpa server](#)

Untuk AWS DMS Tanpa Server, panel navigasi sebelah kiri AWS DMS konsol memiliki opsi baru, replikasi Tanpa Server. Untuk Replikasi Tanpa Server, Anda menentukan Replikasi alih-alih jenis instans replikasi atau tugas untuk menentukan replikasi. Selain itu, Anda menentukan unit kapasitas DMS maksimum dan minimum (DCU) yang Anda ingin DMS berikan untuk replikasi. DCU adalah

2GB RAM. AWS DMS menagih akun Anda untuk setiap DCU yang saat ini digunakan replikasi Anda. Untuk informasi tentang AWS DMS harga, lihat [harga AWS Database Migration Service](#).

AWS DMS kemudian secara otomatis menyediakan sumber daya replikasi berdasarkan pemetaan tabel Anda dan perkiraan ukuran beban kerja Anda. Unit kapasitas ini adalah nilai dalam kisaran nilai unit kapasitas minimum dan maksimum yang Anda tentukan.

Versi Mesin yang Didukung

Dengan AWS DMS Tanpa Server, Anda tidak perlu memilih dan mengelola versi mesin, karena layanan menangani pengaturan itu. AWS DMS Tanpa server mendukung sumber-sumber berikut:

- Microsoft SQL Server
- Database yang kompatibel dengan PostgreSQL
- Database yang kompatibel dengan MySQL
- MariaDB
- Oracle
- IBM Db2

AWS DMS Tanpa server mendukung target berikut:

- Microsoft SQL Server
- PostgreSQL
- Database yang kompatibel dengan MySQL
- Oracle
- Amazon S3
- Amazon Redshift
- Amazon DynamoDB
- Amazon Kinesis Data Streams
- Amazon Managed Streaming untuk Apache Kafka
- OpenSearch Layanan Amazon
- Amazon DocumentDB (dengan kompatibilitas MongoDB)
- Amazon Neptune

Sebagai bagian dari AWS DMS Tanpa Server, Anda memiliki akses ke perintah konsol yang memungkinkan Anda membuat, mengonfigurasi, memulai, dan mengelola replikasi tanpa AWS DMS server. Untuk menjalankan perintah ini menggunakan bagian replikasi Tanpa Server pada konsol, Anda perlu melakukan salah satu hal berikut:

- Siapkan kebijakan AWS Identity and Access Management (IAM) baru dan peran IAM untuk melampirkan kebijakan tersebut.
- Gunakan AWS CloudFormation template untuk menyediakan akses yang Anda butuhkan.

AWS DMS Tanpa server memerlukan peran terkait layanan (SLR) untuk ada di akun Anda. AWS DMS mengelola pembuatan dan penggunaan peran ini. Untuk informasi lebih lanjut tentang memastikan bahwa Anda memiliki SLR yang diperlukan, lihat [Peran terkait layanan untuk Tanpa Server AWS DMS](#).

Membuat replikasi tanpa server

Untuk membuat replikasi tanpa server antara dua AWS DMS titik akhir yang ada, lakukan hal berikut. Untuk informasi tentang membuat AWS DMS titik akhir, lihat [Membuat titik akhir sumber dan target](#).

Membuat replikasi tanpa server

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Pada panel navigasi, pilih Replikasi tanpa server, lalu pilih Buat replikasi.
3. Pada halaman Buat replikasi, tentukan konfigurasi replikasi tanpa server Anda:

Opsi	Tindakan
Nama	Masukkan nama untuk mengidentifikasi replikasi, seperti DMS-replication .
Nama Sumber Daya Amazon Deskriptif (ARN) - Opsional	Biarkan kosong.
Titik akhir basis data sumber	Pilih titik akhir yang ada di akun Anda. Perhatikan bahwa AWS DMS Tanpa Server hanya mendukung

Opsi	Tindakan
	subset dari tipe endpoint yang didukung standar. AWS DMS
Titik akhir basis data target	Pilih titik akhir yang ada di akun Anda. Perhatikan bahwa AWS DMS Tanpa Server hanya mendukung subset dari tipe endpoint yang didukung standar. AWS DMS
Jenis replikasi	<p>Pilih jenis replikasi berdasarkan kebutuhan Anda:</p> <ul style="list-style-type: none"> • Beban penuh: AWS DMS memigrasikan data yang ada saja. • Full load and change data capture (CDC): AWS DMS memigrasikan data yang ada dan perubahan yang terjadi selama replikasi. • Ubah pengambilan data (CDC): AWS DMS hanya memigrasikan perubahan yang terjadi setelah Anda memulai replikasi.

Di bagian Pengaturan, atur pengaturan yang diperlukan replikasi Anda.

Di bagian Pemetaan tabel, atur pemetaan tabel untuk menentukan aturan untuk memilih dan memfilter data yang Anda replikasi. Sebelum Anda menentukan pemetaan, pastikan bahwa Anda meninjau bagian dokumentasi dalam pemetaan jenis data untuk basis data sumber dan target Anda.

Di bagian Pengaturan komputasi, atur pengaturan berikut. Untuk informasi tentang pengaturan Compute Config, lihat. [Konfigurasi Komputasi](#)

Opsi	Tindakan
VPC	Pilih VPC yang ada.
Grup subnet	Pilih grup subnet yang ada.

Opsi	Tindakan
Grup keamanan VPC	Pilih default jika belum dipilih.
AWS Kunci KMS	Tinggalkan (Default) aws/dms dipilih.
Deployment	Tinggalkan apa adanya.
Zona Ketersediaan	Tinggalkan apa adanya.
Unit kapasitas DMS minimum (DCU) - (Opsional)	Biarkan kosong untuk menggunakan nilai default 1 DCU.
Unit kapasitas DMS maksimum (DCU)	Pilih 16 DCU.

Biarkan pengaturan Pemeliharaan apa adanya.

4. Pilih Buat replikasi.

AWS DMS membuat replikasi tanpa server untuk melakukan migrasi Anda.

Memodifikasi replikasi AWS DMS tanpa server

Untuk mengubah konfigurasi replikasi Anda, gunakan `modify-replication-config` tindakan. Anda hanya dapat memodifikasi konfigurasi AWS DMS replikasi yang ada di `CREATED`, `STOPPED`, atau `FAILED` status.

Untuk memodifikasi konfigurasi replikasi tanpa server dengan menggunakan AWS Management Console

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Di panel navigasi, pilih Replikasi tanpa server.
3. Pilih replikasi yang ingin Anda modifikasi. Tabel berikut menjelaskan modifikasi yang dapat Anda buat berdasarkan keadaan replikasi saat ini.

Pengaturan	Deskripsi	Negara yang Diizinkan
Nama	Anda dapat mengubah nama replikasi . Masukkan nama untuk replikasi yang berisi 8 hingga 16 karakter ASCII yang dapat dicetak (tidak termasuk /, ", dan @). Nama harus unik untuk akun Anda untuk Wilayah AWS yang Anda pilih. Anda dapat memilih untuk menambahkan beberapa detail ke nama, seperti menyertakan AWS Wilayah dan tugas yang sedang Anda lakukan, misalnya: west2-mysql2mysql-config1 .	Replicati onState adalah CREATED, STOP atau FAILED.
Titik akhir basis data sumber	Pilih titik akhir sumber baru yang ada sebagai sumber replikasi.	Replicati onState adalah CREATED, atau FAILED kapan Provision State null.
Titik akhir basis data target	Pilih titik akhir target baru yang ada sebagai target untuk replikasi.	Replicati onState adalah CREATED, atau FAILED kapan Provision State null.
Jenis replikasi	Anda dapat memodifikasi jenis replikasi tanpa server.	Replicati onState adalah CREATED, atau FAILED kapan Provision State null.

Pengaturan	Deskripsi	Negara yang Diizinkan
Pengaturan Replikasi	Anda dapat memodifikasi pengaturan replikasi, termasuk mode persiapan tabel target, apakah akan menyertakan kolom LOB dalam replikasi, ukuran LOB maksimum, validasi, dan logging. Untuk informasi selengkapnya, lihat Pengaturan tugas .	Replicati onState adalahCREATED,STOP atauFAILED.
Pemetaan tabel	Anda dapat mengubah setelan pemetaan tabel untuk replikasi tanpa server, termasuk aturan pemilihan dan aturan transformasi. Untuk informasi selengkapnya, lihat Pemetaan tabel .	Replicati onState adalahCREATED,STOP atauFAILED.

Pengaturan	Deskripsi	Negara yang Diizinkan
Konfigurasi komputasi	Anda dapat mengubah pengaturan konfigurasi komputasi untuk replikasi tanpa server, termasuk pengaturan jaringan, pengaturan penskalaan, dan pengaturan pemeliharaan. Untuk informasi tentang pengaturan Compute Config, lihat. Konfigurasi Komputasi	<ul style="list-style-type: none"> • Anda dapat memodifikasi pengaturan penskalaan, pemeliharaan, dan jaringan berikut <code>ReplicationState</code> saat <code>CREATED</code>, <code>STOPPED</code>, atau <code>FAILED</code>: <ul style="list-style-type: none"> • <code>MinCapacityUnits</code> • <code>MaxCapacityUnits</code> • <code>MultiAZ</code> • <code>PreferredMaintenanceWindow</code> • <code>VpcSecurityGroupIds</code> • Anda dapat mengubah pengaturan jaringan dan keamanan berikut saat <code>ReplicationState</code> ada <code>CREATED</code>, atau <code>FAILED</code> kapan <code>ProvisionState</code> <code>null</code>: <ul style="list-style-type: none"> • <code>AvailabilityZone</code> • <code>DnsNameServers</code> • <code>KmsKeyId</code>

Pengaturan	Deskripsi	Negara yang Diizinkan
		<ul style="list-style-type: none"> • <code>ReplicationSubnetGroupId</code>

Konfigurasi Komputasi

Anda mengonfigurasi penyediaan replikasi menggunakan parameter Compute Config atau bagian konsol. Bidang dalam objek Compute Config mencakup yang berikut ini:

Opsi	Deskripsi
<code>MinCapacityUnits</code>	Ini adalah jumlah minimum Unit Kapasitas DMS (DCU) yang AWS DMS akan disediakan. Ini juga merupakan DCU minimum yang dapat diturunkan oleh penskalaan otomatis.
<code>MaxCapacityUnits</code>	Ini adalah Unit Kapasitas DMS maksimum (DCU) yang AWS DMS dapat disediakan, tergantung pada prediksi kapasitas replikasi Anda. Ini juga merupakan DCU maksimum yang dapat ditingkatkan oleh penskalaan otomatis.
<code>KmsKeyId</code>	Kunci enkripsi yang digunakan untuk mengenkripsi penyimpanan replikasi dan informasi koneksi. Jika Anda memilih <code>aws/dms</code> (Default), AWS DMS gunakan kunci KMS default yang terkait dengan akun Anda dan Wilayah AWS Deskripsi dan nomor akun Anda ditampilkan, bersama dengan ARN kunci. Untuk informasi selengkapnya tentang penggunaan kunci enkripsi, lihat Mengatur kunci enkripsi dan menentukan izin AWS KMS . Untuk tutorial ini, biarkan (Default) <code>aws/dms</code> dipilih.
<code>ReplicationSubnetGroupId</code>	Grup subnet replikasi di VPC pilihan Anda di mana Anda ingin replikasi dibuat. Jika database sumber Anda berada dalam VPC, pilih grup subnet yang berisi database

Opsi	Deskripsi
	sumber sebagai lokasi replikasi Anda. Untuk informasi lebih lanjut tentang grup subnet replikasi, lihat Membuat grup subnet replikasi .
VpcSecurityGroupIds	Instans replikasi dibuat dalam VPC. Jika database sumber Anda berada di VPC, pilih grup keamanan VPC yang menyediakan akses ke instans DB tempat database berada.
PreferredMaintenanceWindow	Parameter ini mendefinisikan rentang waktu mingguan di mana pemeliharaan sistem dapat terjadi, dalam Waktu Terkoordinasi Universal (UTC). Defaultnya adalah jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam per Wilayah AWS, terjadi pada hari acak dalam seminggu.
MultiA	Menyetel parameter opsional ini akan membuat replika siaga replikasi Anda di Availability Zone lain untuk dukungan failover. Jika Anda bermaksud menggunakan change data capture (CDC) atau replikasi yang sedang berlangsung, kami sarankan Anda mengaktifkan opsi ini.

Memahami penskalaan otomatis tanpa server AWS DMS

Setelah Anda menyediakan replikasi dan berada dalam RUNNING status, AWS DMS layanan mengelola kapasitas sumber daya yang mendasarinya untuk beradaptasi dengan perubahan beban kerja. Manajemen ini menskalakan sumber daya replikasi berdasarkan pengaturan replikasi berikut:

- MinCapacityUnits
- MaxCapacityUnits

Replikasi meningkat setelah periode melebihi ambang batas pemanfaatan atas, dan turun ketika pemanfaatan kapasitas di bawah ambang batas pemanfaatan kapasitas minimum untuk periode yang lebih lama.

Menyetel penskalaan otomatis tanpa server AWS DMS

Untuk menyetel parameter penskalaan otomatis replikasi, sebaiknya Anda menyetel `MaxCapacityUnits` ke nilai maksimum, dan biarkan AWS DMS mengelola penyediaan sumber daya. Disarankan agar Anda memilih pengaturan kapasitas maksimum DCU terbesar untuk memberikan manfaat terbesar dari auto-scaling, untuk mengakomodasi lonjakan volume transaksi. Kalkulator harga menunjukkan biaya bulanan maksimum jika replikasi Anda terus menggunakan DCU maksimum. DCU maksimum tidak mewakili biaya aktual, karena Anda hanya membayar kapasitas yang digunakan.

Jika replikasi Anda tidak menggunakan sumber dayanya pada kapasitas penuh, secara bertahap AWS DMS akan menghentikan penyediaan sumber daya untuk menghemat biaya Anda. Namun, karena penyediaan dan penonaktifan sumber daya membutuhkan waktu, sebaiknya Anda menyetel `MinCapacityUnits` setelah Anda ke nilai yang dapat menangani lonjakan mendadak yang Anda harapkan dalam beban kerja replikasi Anda. Ini akan menjaga replikasi Anda dari kekurangan penyediaan sementara menyediakan sumber daya untuk tingkat beban kerja yang AWS DMS lebih tinggi.

Jika Anda kurang menyediakan replikasi Anda dengan pengaturan kapasitas maksimum yang terlalu rendah untuk kebutuhan data, atau kapasitas minimum yang terlalu rendah untuk menangani lonjakan mendadak dalam beban kerja replikasi Anda, Anda mungkin melihat metrik Anda secara konsisten pada nilai maksimumnya. `CapacityUtilization` ini dapat menyebabkan replikasi Anda gagal. Jika replikasi Anda gagal karena sumber daya yang kurang disediakan, AWS DMS buat out-of-memory peristiwa di log replikasi Anda. Jika out-of-memory kondisi terjadi karena lonjakan tiba-tiba pada beban kerja replikasi Anda, replikasi akan diskalakan secara otomatis dan dimulai ulang.

Memantau AWS DMS replikasi tanpa server

AWS menyediakan beberapa alat untuk memantau replikasi AWS DMS tanpa server Anda, dan menanggapi potensi insiden:

- [AWS DMS metrik replikasi tanpa server](#)
- [AWS DMS log replikasi tanpa server](#)

AWS DMS metrik replikasi tanpa server

Pemantauan replikasi tanpa server mencakup CloudWatch metrik Amazon untuk statistik berikut. Statistik ini dikelompokkan berdasarkan setiap replikasi tanpa server.

Metrik	Unit	Deskripsi
CapacityUtilization	Persen	Persentase memori yang digunakan oleh replikasi tanpa server
CDC IncomingChanges	Persen	Jumlah total peristiwa perubahan di a point-in-time yang menunggu untuk diterapkan ke target. Perhatikan bahwa ini tidak sama dengan ukuran nilai perubahan transaksi dari titik akhir sumber. Sejumlah besar untuk metrik ini AWS DMS biasanya menunjukkan tidak dapat menerapkan perubahan yang ditangkap pada waktu yang tepat, sehingga menyebabkan latensi target yang tinggi.
CDC LatencySource	Detik	<p>Selang waktu, dalam detik, antara tindakan terakhir yang ditangkap dari titik akhir sumber dan stempel waktu sistem saat ini dari instans AWS DMS . CDC LatencySource mewakili latensi antara sumber dan contoh replikasi. CDC tinggi LatencySource berarti proses menangkap perubahan dari sumber tertunda. Untuk mengidentifikasi latensi dalam replikasi yang sedang berlangsung, Anda dapat melihat metrik ini bersama dengan CDC. LatencyTarget Jika CDC LatencySource dan CDC LatencyTarget tinggi, selidiki LatencySource CDC terlebih dahulu.</p> <p>CDC LatencySource bisa 0 ketika tidak ada jeda replikasi antara sumber dan replikasi. CDC juga LatencySource dapat menjadi nol ketika replikasi mencoba untuk membaca peristiwa berikutnya dalam log transaksi sumber dan tidak ada peristiwa baru dibandingkan dengan terakhir kali dibaca dari sumber. Ketika ini terjadi, replikasi mengatur ulang CDC ke 0LatencySource .</p>
CDC LatencyTarget	Detik	Kesenjangan, dalam detik, antara tempel waktu tindakan pertama yang menunggu untuk melakukan

Metrik	Unit	Deskripsi
		<p>pada target dan stempel waktu saat ini dari instans AWS DMS . Latensi target adalah perbedaan antara waktu server instance replikasi dan id peristiwa tertua yang belum dikonfirmasi yang diteruskan ke komponen target. Dengan kata lain, latensi target adalah perbedaan stempel waktu antara instance replikasi dan peristiwa tertua yang diterapkan tetapi belum dikonfirmasi oleh titik akhir TRG (99%). Ketika CDC LatencyTarget tinggi, ini menunjukkan proses penerapan peristiwa perubahan ke target tertunda. Untuk mengidentifikasi latensi dalam replikasi yang sedang berlangsung, Anda dapat melihat metrik ini bersama dengan CDC.LatencySource. Jika CDC LatencyTarget tinggi tetapi CDC LatencySource tidak tinggi, selidiki apakah:</p> <ul style="list-style-type: none"> • Tidak ada kunci primer atau indeks yang berada di target • Sumber daya kemacetan terjadi dalam instans target atau replikasi • Masalah jaringan berada di antara replikasi dan target
CDC ThroughputBandwidthTarget	KB/detik	Data keluar ditransmisikan untuk target di KB per detik. CDC ThroughputBandwidth mencatat data keluar yang dikirimkan pada titik pengambilan sampel. Jika tidak ada lalu lintas jaringan yang ditemukan, nilainya nol. Karena CDC tidak mengeluarkan transaksi yang berlangsung lama, lalu lintas jaringan mungkin tidak memiliki catatan.
CDC ThroughputRowsSource	Baris/detik	Perubahan yang masuk dari sumber dalam baris per detik.

Metrik	Unit	Deskripsi
CDC Throughput tRowsTarget	Baris/detik	Perubahan keluar untuk target dalam baris per detik.
FullLoadThroughput BandwidthTarget	KB/detik	Data keluar ditransmisikan dari beban penuh untuk target di KB per detik.
FullLoadThroughput RowsTarget	Baris/detik	Perubahan keluar dari beban penuh untuk target dalam baris per detik.

AWS DMS log replikasi tanpa server

Anda dapat menggunakan Amazon CloudWatch untuk mencatat informasi replikasi selama proses AWS DMS migrasi. Anda mengaktifkan pencatatan saat memilih pengaturan replikasi.

Replikasi tanpa server mengunggah log status ke CloudWatch akun Anda untuk meningkatkan visibilitas ke dalam kemajuan replikasi, dan untuk membantu pemecahan masalah.

AWS DMS mengunggah log yang ditautkan tanpa server ke grup log khusus dengan awalan `dms-serverless-replication-<your replication config resource ID>`. Dalam grup log ini, ada aliran log yang disebut `dms-serverless-replication-orchestrator-<your replication config resource ID>`. Aliran log ini melaporkan status replikasi replikasi Anda, dan pesan terkait yang memberikan detail lebih lanjut tentang pekerjaan yang dilakukannya pada tahap ini. Untuk contoh entri log, lihat [Contoh log replikasi tanpa server](#) berikut.

Note

AWS DMS tidak membuat grup log atau streaming sampai Anda menjalankan replikasi. AWS DMS tidak membuat grup log atau streaming jika Anda hanya membuat replikasi.

Untuk melihat log replikasi yang dijalankan, ikuti langkah-langkah berikut:

1. Buka AWS DMS konsol, dan pilih Replikasi tanpa server dari panel navigasi. Dialog replikasi Tanpa Server muncul.

2. Pergi ke bagian Konfigurasi dan pilih Lihat log tanpa server di kolom Umum. Grup CloudWatch log terbuka.
3. Temukan bagian log tugas Migrasi dan pilih Lihat CloudWatch Log.

Jika replikasi Anda gagal, AWS DMS buat entri log dengan status replikasi `failed`, dan pesan yang menjelaskan alasan kegagalan tersebut. Anda harus memeriksa CloudWatch log Anda sebagai langkah pertama dalam memecahkan masalah replikasi yang gagal.

Note

Seperti halnya AWS DMS Classic, Anda memiliki opsi untuk mengaktifkan logging yang lebih terperinci tentang kemajuan migrasi data itu sendiri; yaitu, log yang dipancarkan oleh tugas replikasi yang mendasarinya. Anda dapat mengaktifkan log ini di pengaturan replikasi Anda dengan menyetel `EnableLogging` di `Logging` bidang ke `true`, seperti pada contoh JSON berikut:

```
{
  "Logging": {
    "EnableLogging": true
  }
}
```

Jika Anda mengaktifkan log ini, mereka hanya mulai muncul selama `running` tahap replikasi tanpa server Anda. Mereka akan muncul di bawah grup log yang sama dengan aliran log sebelumnya, tetapi akan berada di bawah aliran log baru `barudms-serverless-serv-res-id-{unique identifier}`. Lihat bagian berikut untuk informasi tentang cara menafsirkan log replikasi tanpa server.

Contoh log replikasi tanpa server

Bagian ini mencakup contoh entri log untuk replikasi tanpa server.

Contoh: Replikasi mulai

Saat Anda menjalankan replikasi tanpa server, AWS DMS buat entri log yang mirip dengan yang berikut ini:

```
{'replication_state':'initializing', 'message': 'Initializing the replication workflow.'}
```

Contoh: Kegagalan replikasi

Jika salah satu titik akhir replikasi tidak dikonfigurasi dengan benar, AWS DMS buat entri log yang mirip dengan berikut ini:

```
{'replication_state':'failed', 'message': 'Test connection failed for endpoint X.', 'failure_message': 'X'}
```

Jika Anda melihat pesan ini di log Anda setelah kegagalan, pastikan titik akhir yang ditentukan sehat dan dikonfigurasi dengan benar.

AWS DMS Batasan tanpa server

AWS DMS Tanpa server memiliki batasan sebagai berikut:

- Anda hanya dapat memodifikasi konfigurasi AWS DMS replikasi yang ada diCREATED,STOPPED, atau FAILED status. Untuk detail tentang pengaturan mana yang dapat Anda ubah dalam kondisi apa, lihat[Memodifikasi replikasi AWS DMS tanpa server](#).
- Anda hanya dapat menghapus konfigurasi AWS DMS replikasi yang ada diSTOPPED, atau FAILED status.
- Penyimpanan statis 100GB yang dialokasikan tersedia untuk replikasi. Jika replikasi Anda menggunakan lebih banyak memori daripada ini, karena persyaratan seperti transaksi yang berjalan lama atau caching, kami sarankan Anda mempartisi beban kerja Anda menjadi replikasi tanpa server yang terpisah. Anda dapat mempartisi beban kerja Anda berdasarkan tabel, atau dengan persyaratan, seperti dengan menempatkan semua replikasi yang melibatkan LOB ke dalam replikasi tanpa server yang terpisah.
- Tidak seperti contoh replikasi, replikasi AWS DMS tanpa server tidak memiliki alamat IP publik untuk tugas-tugas manajemen. Anda mengelola replikasi tanpa server menggunakan konsol.
- Rilis AWS DMS tanpa server ini tidak mendukung semua jenis titik akhir sumber dan target yang AWS DMS didukung standar. Untuk daftar jenis mesin yang didukung, lihat[AWS DMS Komponen tanpa server](#).
- Replikasi tanpa server perlu mengakses dependensi dengan menggunakan titik akhir VPC. Anda harus menggunakan titik akhir VPC untuk mengakses jenis titik akhir berikut:

- Amazon S3
- Amazon Kinesis
- AWS Secrets Manager
- Amazon DynamoDB
- Amazon Redshift
- OpenSearch Layanan Amazon

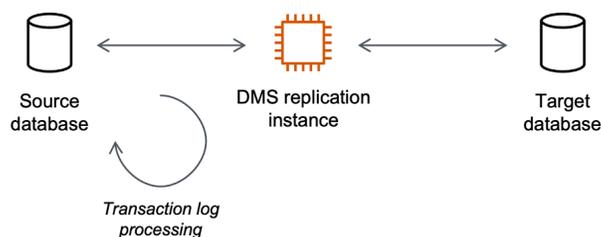
Untuk informasi tentang pengaturan titik akhir VPC, lihat. [Mengkonfigurasi titik akhir VPC sebagai AWS Sumber DMS dan titik akhir target](#)

- AWS DMS tanpa server tidak mendukung tampilan dengan aturan pemilihan dan transformasi.
- AWS DMS tanpa server tidak mendukung penggunaan kunci yang dikelola AWS pelanggan. AWS DMS tanpa server hanya mendukung penggunaan kunci DMS default. Untuk informasi selengkapnya, lihat [Perlindungan data di AWS Database Migration Service](#).
- DMS Serverless tidak mendukung koneksi SSL untuk titik akhir DB2.

Bekerja dengan instance AWS DMS replikasi

Saat Anda membuat instance AWS DMS replikasi, AWS DMS buat instans Amazon EC2 di virtual private cloud (VPC) berdasarkan layanan Amazon VPC. Anda menggunakan instans replikasi ini untuk melakukan migrasi basis data Anda. Dengan menggunakan instans replikasi, Anda bisa mendapatkan ketersediaan tinggi dan dukungan failover dengan deployment Multi-AZ saat Anda memilih opsi Multi-AZ.

Dalam penerapan Multi-AZ, AWS DMS secara otomatis menyediakan dan memelihara replika siaga sinkron dari instance replikasi di Availability Zone yang berbeda. Instans replikasi primer direplikasi secara sinkron di seluruh Availability Zone ke replika siaga. Pendekatan tersebut akan memberikan redundansi data, menghilangkan I/O freeze, dan meminimalkan lonjakan latensi.



AWS DMS menggunakan contoh replikasi untuk terhubung ke penyimpanan data sumber Anda, membaca data sumber, dan memformat data untuk konsumsi oleh penyimpanan data target. Instans replikasi juga memuat data ke penyimpanan data target. Sebagian besar pemrosesan ini terjadi di memori. Namun, transaksi Large mungkin memerlukan beberapa buffering pada disk. Transaksi ter-cache dan berkas log juga ditulis ke disk.

Anda dapat membuat instance AWS DMS replikasi di AWS Wilayah berikut.

Nama Wilayah	Wilayah	Titik Akhir	Protokol
AS Timur (Ohio)	us-east-2	dms.us-east-2.amazonaws.com	HTTPS
		dms-fips.us-east-2.amazonaws.com	HTTPS
AS Timur (Virginia Utara)	us-east-1	dms.us-east-1.amazonaws.com	HTTPS
		dms-fips.us-east-1.amazonaws.com	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
AS Barat (California Utara)	us-west-1	dms.us-west-1.amazonaws.com	HTTPS
		dms-fips.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	dms.us-west-2.amazonaws.com	HTTPS
		dms-fips.us-west-2.amazonaws.com	HTTPS
Africa (Cape Town)	af-south-1	dms.af-south-1.amazonaws.com	HTTPS
Asia Pasifik (Hong Kong)	ap-east-1	dms.ap-east-1.amazonaws.com	HTTPS
Asia Pasifik (Hyderabad)	ap-south-2	dms.ap-south-2.amazonaws.com	HTTPS
Asia Pasifik (Jakarta)	ap-southeast-3	dms.ap-southeast-3.amazonaws.com	HTTPS
Asia Pasifik (Melbourne)	ap-southeast-4	dms.ap-southeast-4.amazonaws.com	HTTPS
Asia Pasifik (Mumbai)	ap-south-1	dms.ap-south-1.amazonaws.com	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Asia Pasifik (Osaka)	ap-northeast-3	dms.ap-northeast-3.amazonaws.com	HTTPS
Asia Pasifik (Seoul)	ap-northeast-2	dms.ap-northeast-2.amazonaws.com	HTTPS
Asia Pasifik (Singapura)	ap-southeast-1	dms.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	dms.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	dms.ap-northeast-1.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	dms.ca-central-1.amazonaws.com	HTTPS
Kanada Barat (Calgary)	ca-west-1	dms.ca-west-1.amazonaws.com	HTTPS
Eropa (Frankfurt)	eu-central-1	dms.eu-central-1.amazonaws.com	HTTPS
Eropa (Irlandia)	eu-west-1	dms.eu-west-1.amazonaws.com	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Europe (London)	eu-west-2	dms.eu-west-2.amazonaws.com	HTTPS
Eropa (Milan)	eu-south-1	dms.eu-south-1.amazonaws.com	HTTPS
Eropa (Paris)	eu-west-3	dms.eu-west-3.amazonaws.com	HTTPS
Eropa (Spanyol)	eu-south-2	dms.eu-south-2.amazonaws.com	HTTPS
Eropa (Stockholm)	eu-north-1	dms.eu-north-1.amazonaws.com	HTTPS
Eropa (Zürich)	eu-central-2	dms.eu-central-2.amazonaws.com	HTTPS
Israel (Tel Aviv)	il-central-1	dms.il-central-1.amazonaws.com	HTTPS
Timur Tengah (Bahrain)	me-south-1	dms.me-south-1.amazonaws.com	HTTPS
Timur Tengah (UAE)	me-central-1	dms.me-central-1.amazonaws.com	HTTPS
Amerika Selatan (Sao Paulo)	sa-east-1	dms.sa-east-1.amazonaws.com	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol	
AWS GovCloud (AS-Timur)	us-gov-east-1	dms.us-gov-east-1.amazonaws.com	HTTPS	
AWS GovCloud (AS-Barat)	us-gov-west-1	dms.us-gov-west-1.amazonaws.com	HTTPS	

AWS DMS mendukung AWS Wilayah khusus yang disebut AWS GovCloud (US) yang dirancang untuk memungkinkan lembaga pemerintah AS dan pelanggan memindahkan beban kerja sensitif ke cloud. AWS GovCloud (US) menangani persyaratan peraturan dan kepatuhan khusus pemerintah AS. Untuk informasi lebih lanjut tentang AWS GovCloud (US), lihat [Apa itu AWS GovCloud \(AS\)?](#)

Di bawah ini, Anda bisa mendapatkan detail lebih lanjut tentang instans replikasi.

Topik

- [Memilih instans replikasi AWS DMS yang tepat untuk migrasi Anda](#)
- [Memilih ukuran terbaik untuk contoh replikasi](#)
- [Bekerja dengan versi mesin replikasi](#)
- [Instans replikasi publik dan privat](#)
- [Alamat IP dan jenis jaringan](#)
- [Menyiapkan jaringan untuk instans replikasi](#)
- [Mengatur kunci enkripsi untuk instans replikasi](#)
- [Membuat instans replikasi](#)
- [Mengubah instans replikasi](#)
- [Mem-boot ulang instans replikasi](#)
- [Menghapus instans replikasi](#)
- [Bekerja dengan jendela pemeliharaan AWS DMS](#)

Memilih instans replikasi AWS DMS yang tepat untuk migrasi Anda

AWS DMS membuat instance replikasi pada instans Amazon EC2. AWS DMS saat ini mendukung kelas instans Amazon EC2, T3, C4, C5, C6i, R4, R5 dan R6i Amazon EC2 untuk instance replikasi:

- Instans T2 adalah instans performa yang dapat dilonjakkan yang memberikan performa CPU tingkat baseline dengan kemampuan untuk melonjak di atas baseline. Performa dasar dan kemampuan melonjak diatur oleh kredit CPU. Instans T2 menerima kredit CPU secara terus-menerus pada tingkat yang sudah ditetapkan, tergantung pada ukuran instans. Instans tersebut mengumpulkan kredit CPU pada saat diam dan mengonsumsi kredit CPU pada saat aktif.

Instans T2 adalah pilihan yang baik untuk berbagai beban kerja tujuan umum. Beban kerja tersebut termasuk layanan mikro, aplikasi interaktif latensi rendah, basis data kecil dan menengah, desktop virtual, lingkungan pengembangan, pembuatan, dan panggung, repositori kode, dan prototipe produk.

- Instans T3 adalah tipe instans tujuan umum yang dapat dilonjakkan generasi berikutnya. Tipe ini menyediakan performa CPU tingkat baseline dengan kemampuan untuk meningkatkan penggunaan CPU kapan saja selama yang diperlukan. Instans T3 menawarkan keseimbangan komputasi, memori, dan sumber daya jaringan serta dirancang untuk aplikasi dengan penggunaan CPU moderat yang mengalami lonjakan sementara pada saat digunakan. Instans T3 mengumpulkan kredit CPU ketika beban kerja beroperasi di bawah ambang batas baseline. Setiap kredit CPU yang diperoleh memberikan kesempatan pada instans T3 untuk melonjak dengan performa satu core CPU selama satu menit bila diperlukan.

Instans T3 dapat melonjak kapan saja selama yang diperlukan dalam mode `unlimited`. Untuk informasi lebih lanjut tentang mode `unlimited`, lihat [Bekerja dengan mode tak terbatas untuk instans performa yang dapat dilonjakkan](#).

- Instans C4 dioptimalkan untuk beban kerja intensif komputasi dan memberikan performa tinggi yang sangat hemat biaya dengan rasio harga per komputasi yang rendah. Mereka mencapai kinerja paket per detik (PPS) yang jauh lebih tinggi, jitter jaringan yang lebih rendah, dan latensi jaringan yang lebih rendah. AWS DMS Bisa juga intensif CPU, terutama saat melakukan migrasi dan replikasi heterogen seperti bermigrasi dari Oracle ke PostgreSQL. Instans C4 dapat menjadi pilihan yang bagus untuk situasi ini.
- Instans C5 adalah tipe instans generasi berikutnya yang digunakan untuk memberikan performa tinggi yang hemat biaya dengan rasio harga per komputasi yang rendah untuk menjalankan beban kerja intensif komputasi lanjutan. Beban kerja tersebut termasuk server web performa tinggi, komputasi performa tinggi (HPC), pemrosesan batch, penayangan iklan, game multipemain yang

dapat diskalakan, dan pengkodean video. Instans C5 beban kerja lainnya cocok untuk mencakup pemodelan ilmiah, analitik terdistribusi, dan inferensi mesin dan deep learning. Instans C5 tersedia dengan pilihan prosesor dari Intel dan AMD.

- Instans C6i menawarkan kinerja harga komputasi hingga 15% lebih baik dibandingkan instans Gen5 yang sebanding untuk berbagai macam beban kerja, dan enkripsi memori yang selalu aktif. Instans C6i sangat cocok untuk beban kerja intensif komputasi seperti pemrosesan batch, analitik terdistribusi, komputasi kinerja tinggi (HPC), penayangan iklan, game multipemain yang sangat skalabel, dan pengkodean video.
- Instans R4 adalah memori yang dioptimalkan untuk beban kerja intensif memori. Migrasi yang sedang berlangsung atau replikasi sistem transaksi throughput tinggi yang menggunakan AWS DMS juga dapat mengonsumsi CPU dan memori dalam jumlah besar. Instans R4 mencakup lebih banyak memori per vCPU dibandingkan tipe instans generasi sebelumnya.
- Instans R5 adalah generasi berikutnya dari tipe instans memori yang dioptimalkan untuk memori untuk Amazon EC2. Instans R5 sangat sesuai untuk aplikasi intensif memori seperti basis data performa tinggi, cache dalam memori skala web terdistribusi, basis data dalam memori berukuran sedang, analitik big data waktu nyata, dan aplikasi korporasi lainnya. Migrasi yang sedang berlangsung atau replikasi sistem transaksi throughput tinggi yang digunakan juga AWS DMS dapat mengonsumsi CPU dan memori dalam jumlah besar.
- Instans R6i menawarkan kinerja harga komputasi hingga 15% lebih baik dibandingkan instans Gen5 yang sebanding untuk berbagai macam beban kerja, dan enkripsi memori yang selalu aktif. Instans R6i adalah SAP Certified dan ideal untuk beban kerja seperti database SQL dan NoSQL, cache dalam memori skala web terdistribusi seperti Memcached dan Redis, database dalam memori seperti SAP HANA, dan analisis data besar real time seperti cluster Hadoop dan Spark.

Setiap instans replikasi memiliki konfigurasi memori dan vCPU tertentu. Tabel berikut menunjukkan konfigurasi untuk setiap tipe instans replikasi. Untuk informasi harga, lihat [AWS Database Migration Service halaman harga layanan](#).

Jenis Instans Replikasi Tujuan Umum

Tipe	vCPU	Memori (GiB)
dms.t2.micro	1	1
dms.t2.small	1	2

Tipe	vCPU	Memori (GiB)
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

Menghitung Jenis Instans Replikasi yang Dioptimalkan

Tipe	vCPU	Memori (GiB)
dms.c4.large	2	3,75
dms.c4.xlarge	4	7,5
dms.c4.2xlarge	8	15
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

Tipe	vCPU	Memori (GiB)
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

Jenis Instans Replikasi yang Dioptimalkan Memori

Tipe	vCPU	Memori (GiB)
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

Tipe	vCPU	Memori (GiB)
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
dms.r6i.24xlarge	96	768
dms.r6i.32xlarge	128	1024

Tabel di atas mencantumkan semua jenis contoh AWS DMS replikasi, tetapi jenis yang tersedia di wilayah Anda mungkin berbeda. Untuk melihat jenis instance replikasi yang tersedia di wilayah Anda, Anda dapat menjalankan [AWS CLI](#) perintah berikut:

```
aws dms describe-orderable-replication-instances --region your_region_name
```

Topik

- [Menentukan kelas instans yang digunakan](#)

- [Bekerja dengan mode tak terbatas untuk instans performa yang dapat dilonjakkan](#)

Menentukan kelas instans yang digunakan

Untuk membantu menentukan kelas instance replikasi mana yang paling cocok untuk Anda, mari kita lihat proses change data capture (CDC) yang AWS DMS digunakan.

Mari kita asumsikan bahwa Anda sedang menjalankan beban penuh ditambah dengan tugas CDC (beban massal ditambah dengan replikasi yang sedang berlangsung). Pada kasus ini, tugas tersebut memiliki repositori SQLite sendiri untuk menyimpan metadata dan informasi lainnya. Sebelum AWS DMS memulai beban penuh, langkah-langkah ini terjadi:

- AWS DMS mulai menangkap perubahan untuk tabel yang dimigrasikan dari log transaksi mesin sumber (kami menyebutnya perubahan cache). Setelah beban penuh dilakukan, perubahan ter-cache tersebut dikumpulkan dan diterapkan pada target. Tergantung pada volume perubahan ter-cache, perubahan tersebut dapat langsung diterapkan dari memori, di mana mereka pertama kali dikumpulkan, hingga ambang batas yang ditetapkan. Atau mereka dapat diterapkan dari disk, di mana perubahan ditulis ketika perubahan tersebut tidak dapat disimpan di memori.
- Setelah perubahan cache diterapkan, secara default AWS DMS memulai proses penerapan transaksional pada instance target.

Selama fase perubahan cache yang diterapkan dan fase replikasi yang sedang berlangsung, AWS DMS menggunakan dua buffer aliran, masing-masing untuk data masuk dan keluar. AWS DMS juga menggunakan komponen penting yang disebut penyortir, yang merupakan buffer memori lain. Berikut ini adalah dua kegunaan penting dari komponen penyortir (yang memiliki kegunaan lain):

- Penyortir melacak semua transaksi dan memastikan bahwa penyortir hanya meneruskan transaksi yang relevan untuk buffer keluar.
- Penyortir memastikan bahwa transaksi diteruskan dalam urutan perlakuan yang sama seperti pada sumber.

Seperti yang Anda lihat, kami memiliki tiga buffer memori penting dalam arsitektur ini untuk CDC di AWS DMS. Jika salah satu buffer tersebut mengalami tekanan memori, migrasi dapat memiliki masalah performa yang berpotensi menyebabkan kegagalan.

Saat Anda memasukkan beban kerja berat dengan jumlah transaksi per detik (TPS) yang tinggi ke dalam arsitektur ini, Anda dapat menemukan memori tambahan yang disediakan oleh instans R5

dan R6i berguna. Anda dapat menggunakan instans R5 dan R6i untuk menyimpan sejumlah besar transaksi dalam memori dan mencegah masalah tekanan memori selama replikasi yang sedang berlangsung.

Bekerja dengan mode tak terbatas untuk instans performa yang dapat dilonjakkan

Instans performa yang dapat dilonjakkan yang dikonfigurasi sebagai `unlimited`, seperti instans T3, dapat mempertahankan pemakaian CPU yang tinggi untuk jangka waktu apa pun kapan pun diperlukan. Harga instans per jam dapat secara otomatis mencakup semua lonjakan penggunaan CPU. Hal tersebut terjadi jika penggunaan CPU rata-rata dari instans berada pada atau di bawah baseline selama periode 24 jam bergulir atau masa pakai instans, mana saja yang lebih pendek.

Untuk sebagian besar beban kerja tujuan umum, instans yang dikonfigurasi sebagai `unlimited` memberikan performa yang cukup tanpa biaya tambahan. Jika instans berjalan pada pemakaian CPU yang lebih tinggi untuk waktu yang lama, instans dapat melakukannya dengan tarif tambahan tetap per jam vCPU. Untuk informasi tentang harga instans T3, lihat "Kredit CPU T3" di [AWS Database Migration Service](#).

Untuk informasi lebih lanjut tentang mode `unlimited` untuk instans T3, lihat [Mode tak terbatas untuk instans performa yang dapat dilonjakkan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Important

Jika Anda menggunakan instans `dms.t3.micro` di bawah penawaran [Tingkat Gratis AWS](#) dan menggunakannya dalam mode `unlimited`, biaya mungkin berlaku. Secara khusus, biaya mungkin berlaku jika penggunaan rata-rata selama periode 24 jam bergulir melebihi pemanfaatan dasar instans. Untuk informasi lebih lanjut, lihat [Pemanfaatan dasar](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Instans T3 diluncurkan sebagai `unlimited` secara default. Jika penggunaan CPU rata-rata selama periode 24 jam melebihi standar, Anda dikenai biaya untuk kelebihan kredit. Dalam beberapa kasus, Anda mungkin meluncurkan Instans Spot T3 sebagai `unlimited` dan berencana untuk segera menggunakannya dan untuk durasi yang singkat. Jika Anda melakukannya tanpa waktu diam untuk memperoleh kredit CPU, Anda dikenakan biaya untuk kelebihan kredit. Kami menyarankan Anda untuk meluncurkan Instans Spot T3 dalam mode standar untuk menghindari pembayaran biaya yang lebih tinggi. Untuk informasi lebih lanjut,

lihat [Kredit surplus dapat dikenakan biaya](#), [Instans Spot T3](#), dan [Mode standar untuk instans performa yang dapat dilonjakkan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Memilih ukuran terbaik untuk contoh replikasi

Memilih instans replikasi yang sesuai tergantung pada beberapa faktor kasus penggunaan Anda. Untuk membantu memahami bagaimana sumber daya instans replikasi digunakan, lihat diskusi berikut. Ini mencakup skenario umum beban penuh + tugas CDC.

Selama tugas beban penuh, AWS DMS memuat tabel satu per satu. Secara default, delapan tabel dimuat sekaligus. AWS DMS menangkap perubahan yang sedang berlangsung pada sumber selama tugas pemuatan penuh sehingga perubahan dapat diterapkan nanti pada titik akhir target. Perubahan disimpan dalam memori; jika memori yang tersedia habis, perubahan disimpan ke disk. Ketika tugas beban penuh selesai untuk tabel, AWS DMS segera terapkan perubahan cache ke tabel target.

Setelah semua perubahan ter-cache yang tertunda untuk sebuah tabel telah diterapkan, titik akhir target berada dalam keadaan konsisten secara transaksi. Pada titik ini, target sinkron dengan titik akhir sumber sehubungan dengan perubahan cache terakhir. AWS DMS kemudian mulai replikasi yang sedang berlangsung antara sumber dan target. Untuk melakukannya, AWS DMS mengambil operasi perubahan dari log transaksi sumber dan menerapkannya ke target dengan cara yang konsisten secara transaksional. (Proses ini mengasumsikan penerapan yang dioptimalkan batch tidak dipilih). AWS DMS mengalirkan perubahan yang sedang berlangsung melalui memori pada instance replikasi, jika memungkinkan. Jika tidak, AWS DMS tulis perubahan pada disk pada instance replikasi hingga dapat diterapkan pada target.

Anda dapat mengontrol bagaimana instans replikasi menangani pemrosesan perubahan, dan bagaimana memori digunakan dalam proses tersebut. Untuk informasi lebih lanjut tentang cara mengatur pemrosesan perubahan, lihat [Mengubah pengaturan penyetelan pemrosesan](#).

Faktor yang perlu dipertimbangkan

Memori dan ruang disk adalah faktor kunci dalam memilih contoh replikasi yang sesuai untuk kasus penggunaan Anda. Berikut ini, Anda dapat menemukan diskusi tentang karakteristik kasus penggunaan untuk menganalisis untuk memilih contoh replikasi.

- Database dan ukuran tabel

Volume data membantu menentukan konfigurasi tugas untuk mengoptimalkan kinerja beban penuh. Misalnya, untuk dua skema 1 TB, Anda dapat mempartisi tabel menjadi empat tugas 500 GB dan menjalankannya secara paralel. Paralelisme yang mungkin tergantung pada sumber daya CPU yang tersedia dalam contoh replikasi. Itu sebabnya ide yang baik memahami ukuran database dan tabel Anda untuk mengoptimalkan kinerja beban penuh. Ini membantu menentukan jumlah tugas yang mungkin dapat Anda miliki.

- Benda besar

Tipe data yang ada dalam lingkup migrasi Anda dapat memengaruhi kinerja. Khususnya, objek besar (LOB) berdampak pada kinerja dan konsumsi memori. Untuk memigrasikan nilai LOB, AWS DMS lakukan proses dua langkah. Pertama, AWS DMS masukkan baris ke target tanpa nilai LOB. Kedua, AWS DMS memperbarui baris dengan nilai LOB. Ini berdampak pada memori, jadi penting untuk mengidentifikasi kolom LOB di sumber dan menganalisis ukurannya.

- Frekuensi beban dan ukuran transaksi

Frekuensi beban dan transaksi per detik (TPS) mempengaruhi penggunaan memori. Tingginya jumlah aktivitas TPS atau data manipulation language (DHTML) menyebabkan penggunaan memori yang tinggi. Ini terjadi karena DMS menyimpan perubahan sampai diterapkan ke target. Selama CDC, ini menyebabkan pertukaran (menulis ke disk fisik karena luapan memori), yang menyebabkan latensi.

- Tombol tabel dan integritas referensial

Informasi tentang kunci tabel menentukan mode CDC (penerapan batch atau berlaku transaksional) yang Anda gunakan untuk memigrasikan data. Secara umum, penerapan transaksional lebih lambat daripada penerapan batch. Untuk transaksi yang berjalan lama, mungkin ada banyak perubahan untuk bermigrasi. Saat Anda menggunakan aplikasi transaksional, AWS DMS mungkin memerlukan lebih banyak memori untuk menyimpan perubahan dibandingkan dengan penerapan batch. Jika Anda memigrasikan tabel tanpa kunci utama, penerapan batch akan gagal dan tugas DMS pindah ke mode penerapan transaksional. Ketika integritas referensial aktif antara tabel selama CDC, AWS DMS menggunakan penerapan transaksional secara default. Untuk informasi lebih lanjut tentang penerapan batch dibandingkan dengan penerapan transaksional, [lihat Bagaimana cara menggunakan fitur penerapan batch DMS untuk meningkatkan kinerja replikasi CDC?](#)

Gunakan metrik ini untuk menentukan apakah Anda memerlukan instance replikasi untuk dioptimalkan komputasi atau memori dioptimalkan.

Masalah umum

Anda mungkin menghadapi masalah umum berikut yang menyebabkan pertentangan sumber daya pada instance replikasi selama migrasi. Untuk informasi tentang metrik instance replikasi, lihat [Metrik instans replikasi](#)

- Jika memori dalam contoh replikasi menjadi tidak mencukupi, ini menghasilkan penulisan data ke disk. Membaca dari disk dapat menyebabkan latensi, yang dapat Anda hindari dengan mengukur instance replikasi dengan memori yang cukup.
- Ukuran disk yang ditetapkan untuk instance replikasi bisa lebih kecil dari yang dibutuhkan. Ukuran disk digunakan ketika data dalam memori tumpah; itu juga digunakan untuk menyimpan log tugas. IOPS maksimum juga tergantung padanya.
- Menjalankan beberapa tugas atau tugas dengan paralelisme tinggi memengaruhi konsumsi CPU dari instance replikasi. Ini memperlambat pemrosesan tugas dan menghasilkan latensi.

Praktik terbaik

Pertimbangkan dua praktik terbaik yang paling umum ini saat mengukur contoh replikasi. Untuk informasi selengkapnya, lihat [Praktik terbaik AWS Database Migration Service](#).

1. Ukur beban kerja Anda dan pahami apakah itu padat komputer atau intensif memori. Berdasarkan ini, Anda dapat menentukan kelas dan ukuran instance replikasi:
 - AWS DMS memproses LOB dalam memori. Operasi ini membutuhkan cukup banyak memori.
 - Jumlah tugas dan jumlah thread memengaruhi konsumsi CPU. Hindari menggunakan lebih dari delapan `MaxFullLoadSubTasks` selama operasi beban penuh.
2. Tingkatkan ruang disk yang ditetapkan ke instance replikasi saat Anda memiliki beban kerja yang tinggi selama beban penuh. Melakukan hal ini memungkinkan instance replikasi menggunakan IOPS maksimum yang ditetapkan untuk itu.

Pedoman sebelumnya tidak mencakup semua skenario yang mungkin. Penting untuk mempertimbangkan secara spesifik kasus penggunaan khusus Anda saat Anda menentukan ukuran instance replikasi Anda.

Tes sebelumnya menunjukkan CPU dan memori bervariasi dengan beban kerja yang berbeda. Secara khusus, LOB mempengaruhi memori, dan jumlah tugas atau paralelisme mempengaruhi CPU. Setelah migrasi Anda berjalan, pantau CPU, memori yang dapat dibebaskan, penyimpanan

gratis, dan IOPS dari instance replikasi Anda. Berdasarkan data yang Anda kumpulkan, Anda dapat menambah atau mengurangi ukuran instans replikasi Anda sesuai kebutuhan.

Bekerja dengan versi mesin replikasi

Mesin replikasi adalah AWS DMS perangkat lunak inti yang berjalan pada instance replikasi Anda dan melakukan tugas migrasi yang Anda tentukan. AWS secara berkala merilis versi baru dari perangkat lunak mesin AWS DMS replikasi, dengan fitur baru dan peningkatan kinerja. Setiap versi perangkat lunak mesin replikasi memiliki nomor versi sendiri, untuk membedakannya dari versi lain.

Saat Anda meluncurkan instance replikasi baru, ia menjalankan versi AWS DMS mesin terbaru kecuali Anda menentukan sebaliknya. Untuk informasi selengkapnya, lihat [Bekerja dengan instance AWS DMS replikasi](#).

Jika Anda memiliki instance replikasi yang sedang berjalan, Anda dapat memutakhirkannya ke versi mesin yang lebih baru. (AWS DMS tidak mendukung penurunan versi mesin.) Untuk informasi lebih lanjut tentang versi mesin replikasi, lihat [AWS Catatan rilis DMS](#).

Meningkatkan versi mesin menggunakan konsol

Anda dapat memutakhirkan instance AWS DMS replikasi menggunakan file. AWS Management Console

Untuk meningkatkan instans replikasi menggunakan konsol tersebut

1. Buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Di panel navigasi, pilih Instans replikasi.
3. Pilih mesin replikasi Anda, dan kemudian pilih Ubah.
4. Untuk versi Engine, pilih nomor versi yang Anda inginkan, lalu pilih Ubah.

Note

Sebaiknya hentikan semua tugas sebelum memutakhirkan Instance Replikasi. Jika Anda tidak menghentikan tugas, AWS DMS akan menghentikan tugas secara otomatis sebelum upgrade. Jika Anda menghentikan tugas secara manual, Anda harus memulai tugas secara manual setelah peningkatan selesai. Meningkatkan instans replikasi memerlukan waktu beberapa menit. Ketika instans siap, statusnya berubah menjadi tersedia.

Memutakhirkan versi mesin menggunakan AWS CLI

Anda dapat memutakhirkan instance AWS DMS replikasi menggunakan AWS CLI, sebagai berikut.

Untuk memutakhirkan instance replikasi menggunakan AWS CLI

1. Tentukan Amazon Resource Name (ARN) instans replikasi Anda dengan menggunakan perintah berikut.

```
aws dms describe-replication-instances \  
--query "ReplicationInstances[*].  
[ReplicationInstanceIdentifier,ReplicationInstanceArn,ReplicationInstanceClass]"
```

Dalam output, perhatikan ARN untuk instans replikasi yang ingin Anda tingkatkan, misalnya:
`arn:aws:dms:us-east-1:123456789012:rep:6EFQQ06U6EDPRCPKLNPL2SCEEY`

2. Tentukan versi instans replikasi apa saja yang tersedia dengan menggunakan perintah berikut.

```
aws dms describe-orderable-replication-instances \  
--query "OrderableReplicationInstances[*].[ReplicationInstanceClass,EngineVersion]"
```

Dalam output, perhatikan nomor versi mesin atau nomor yang tersedia untuk kelas instans replikasi Anda. Anda akan melihat informasi ini dalam output dari langkah 1.

3. Tingkatkan instans replikasi dengan menggunakan perintah berikut.

```
aws dms modify-replication-instance \  
--replication-instance-arn arn \  
--engine-version n.n.n
```

Ganti *arn* sebelumnya dengan ARN instans replikasi yang sebenarnya dari langkah sebelumnya.

Ganti *n.n.n* dengan nomor versi mesin yang Anda inginkan, misalnya: 3.4.5

Note

Meningkatkan instans replikasi memerlukan waktu beberapa menit. Anda dapat melihat status instans replikasi menggunakan perintah berikut.

```
aws dms describe-replication-instances \
--query "ReplicationInstances[*].
[ReplicationInstanceIdentifier,ReplicationInstanceStatus]"
```

Ketika instans replikasi siap, statusnya berubah ke tersedia.

Instans replikasi publik dan privat

Anda dapat menentukan apakah instance replikasi memiliki alamat IP publik atau pribadi yang digunakan instance untuk terhubung ke basis data sumber dan target.

Instans replikasi privat memiliki alamat IP privat yang tidak dapat Anda akses di luar jaringan replikasi. Anda menggunakan instance pribadi ketika database sumber dan target berada di jaringan yang sama yang terhubung ke virtual private cloud (VPC) dari instance replikasi. Jaringan dapat dihubungkan ke VPC dengan menggunakan virtual private network (VPN) AWS Direct Connect, atau VPC peering.

Koneksi peering VPC adalah koneksi jaringan antara dua VPC. Hal ini memungkinkan routing menggunakan masing-masing alamat IP pribadi VPC seolah-olah mereka berada di jaringan yang sama. Untuk informasi lebih lanjut mengenai peering VPC, lihat [peering VPC](#) dalam Panduan Pengguna Amazon VPC.

Sebuah instance replikasi publik dapat menggunakan grup keamanan VPC dari instance replikasi, dan alamat IP publik instance replikasi atau alamat IP publik gateway NAT. Koneksi tersebut membentuk jaringan yang Anda gunakan untuk migrasi data.

Alamat IP dan jenis jaringan

AWS DMS selalu membuat instance replikasi Anda di Amazon Virtual Private Cloud (VPC). Saat membuat VPC, Anda dapat menentukan alamat IP yang akan digunakan: IPv4 atau IPv6, atau keduanya. Kemudian, ketika Anda membuat atau memodifikasi instance replikasi, Anda dapat menentukan penggunaan protokol alamat IPv4 atau protokol alamat IPv6 menggunakan mode dual-stack.

Alamat IPv4

Saat Anda membuat VPC, Anda dapat menentukan rentang alamat IPv4 untuk VPC dalam bentuk blok Classless Inter-Domain Routing (CIDR), seperti 10.0.0.0/16. Grup subnet mendefinisikan rentang alamat IP di blok CIDR ini. Alamat IP ini bisa bersifat privat atau publik.

Alamat IPv4 privat adalah alamat IP yang tidak dapat dijangkau dengan Internet. Anda dapat menggunakan alamat IPv4 pribadi untuk komunikasi antara instans replikasi dan sumber daya lainnya, seperti instans Amazon EC2, di VPC yang sama. Setiap contoh replikasi memiliki alamat IP pribadi untuk komunikasi di VPC.

Alamat IP publik adalah alamat IPv4 yang dapat dijangkau dari internet. Anda dapat menggunakan alamat publik untuk komunikasi antara contoh replikasi Anda dan sumber daya di internet. Anda mengontrol apakah instance replikasi Anda menerima alamat IP publik.

Mode tumpukan ganda dan alamat IPv6

Bila Anda memiliki sumber daya yang harus berkomunikasi dengan instance replikasi Anda melalui IPv6, gunakan mode dual-stack. Untuk menggunakan mode dual-stack, pastikan bahwa setiap subnet dalam grup subnet yang Anda kaitkan dengan instance replikasi memiliki blok IPv6 CIDR yang terkait dengannya. Anda dapat membuat grup subnet replikasi baru atau memodifikasi grup subnet replikasi yang ada untuk memenuhi persyaratan ini. Setiap alamat IPv6 bersifat unik secara global. Blok IPv6 CIDR untuk VPC Anda secara otomatis ditetapkan dari kumpulan alamat IPv6 Amazon. Anda tidak dapat memilih sendiri rentang tersebut.

DMS menonaktifkan akses Internet Gateway untuk titik akhir IPv6 dari instance replikasi mode dual-stack pribadi. DMS melakukan ini untuk memastikan bahwa titik akhir IPv6 Anda bersifat pribadi dan hanya dapat diakses dari dalam VPC Anda.

Anda dapat menggunakan AWS DMS Konsol untuk membuat atau memodifikasi instance replikasi, dan menentukan mode dual-stack di bagian Jenis jaringan. Gambar berikut menampilkan bagian Jenis jaringan di konsol.

Connectivity and security

Network type - new [Info](#)

To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4

Replication instance with an IPv4 network type that supports IPv4 addressing.

Dual-stack mode

Replication instance with a dual network type that supports both IPv4 and IPv6 addressing.

Referensi

- Untuk informasi mode tentang alamat IPv4 dan IPv6, lihat [Pengalamatan IP](#) di Panduan Pengguna Amazon VPC.
- Untuk informasi selengkapnya tentang membuat instance replikasi menggunakan mode dual-stack, lihat. [Membuat instans replikasi](#)
- Untuk informasi mode tentang memodifikasi instance replikasi, lihat. [Mengubah instans replikasi](#)

Menyiapkan jaringan untuk instans replikasi

AWS DMS selalu membuat instance replikasi dalam VPC berbasis Amazon VPC. Anda menentukan VPC tempat instans replikasi Anda terletak. Anda dapat menggunakan VPC default untuk akun dan AWS Wilayah Anda, atau Anda dapat membuat VPC baru.

Pastikan bahwa antarmuka jaringan elastis yang dialokasikan untuk VPC instans replikasi Anda dikaitkan dengan grup keamanan. Pastikan juga bahwa aturan grup keamanan tersebut membiarkan semua lalu lintas pada semua port meninggalkan (keluar) VPC. Pendekatan ini memungkinkan komunikasi dari instance replikasi ke titik akhir basis data sumber dan target Anda, jika aturan ingress yang benar diaktifkan pada titik akhir. Kami merekomendasikan Anda untuk menggunakan pengaturan default untuk titik akhir, yang memungkinkan jalan keluar pada semua port ke semua alamat.

Titik akhir sumber dan target mengakses instans replikasi yang ada di dalam VPC baik dengan cara menghubungkan ke VPC atau dengan berada di dalam VPC. Titik akhir basis data harus mencakup daftar kontrol akses (ACL) jaringan dan aturan grup keamanan (jika berlaku) yang mengizinkan akses masuk dari instans replikasi. Cara Anda mengatur hal ini tergantung pada konfigurasi jaringan yang Anda gunakan. Anda dapat menggunakan grup keamanan VPC instans replikasi, alamat IP privat atau publik instans replikasi, atau alamat IP publik gateway NAT. Koneksi tersebut membentuk jaringan yang Anda gunakan untuk migrasi data.

Note

Karena alamat IP dapat berubah sebagai akibat dari perubahan infrastruktur yang mendasarinya, kami sarankan Anda menggunakan rentang CIDR VPC, atau merutekan lalu lintas keluar instance replikasi Anda melalui IP Elastis terkait NAT GW. Untuk informasi selengkapnya tentang membuat VPC, termasuk blok CIDR, lihat [Bekerja dengan VPC dan subnet](#) di Panduan Pengguna Amazon Virtual Private Cloud. Untuk informasi selengkapnya

tentang alamat IP Elastis, lihat [Alamat IP Elastis](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Konfigurasi jaringan untuk migrasi basis data

Anda dapat menggunakan beberapa konfigurasi jaringan yang berbeda dengan AWS Database Migration Service. Berikut ini adalah konfigurasi umum untuk jaringan yang digunakan untuk migrasi basis data.

Topik

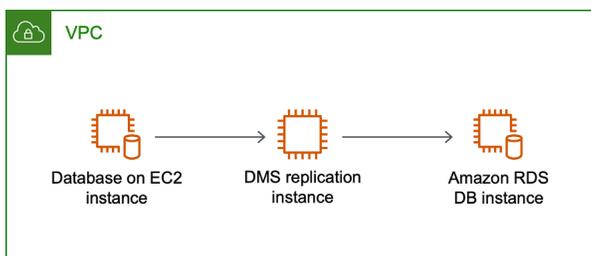
- [Konfigurasi dengan semua komponen migrasi basis data dalam satu VPC](#)
- [Konfigurasi dengan beberapa VPC](#)
- [Konfigurasi dengan VPC bersama](#)
- [Konfigurasi untuk jaringan ke VPC menggunakan AWS Direct Connect atau VPN](#)
- [Konfigurasi untuk jaringan ke VPC menggunakan internet](#)
- [Konfigurasi dengan instans RDS DB tidak dalam VPC ke instance DB di VPC menggunakan ClassicLink](#)

Jika praktis, kami menyarankan Anda membuat instance replikasi DMS di Wilayah yang sama dengan titik akhir target Anda, dan di VPC atau subnet yang sama dengan titik akhir target Anda.

Konfigurasi dengan semua komponen migrasi basis data dalam satu VPC

Jaringan yang paling sederhana untuk migrasi basis data adalah jika titik akhir sumber, instans replikasi, dan titik akhir target berada di VPC yang sama. Konfigurasi ini termasuk baik jika titik akhir sumber dan target Anda berada di instans DB Amazon RDS atau instans Amazon EC2.

Ilustrasi berikut menunjukkan konfigurasi di mana basis data pada instans Amazon EC2 terhubung ke instans replikasi dan data dimigrasi ke instans DB Amazon RDS.



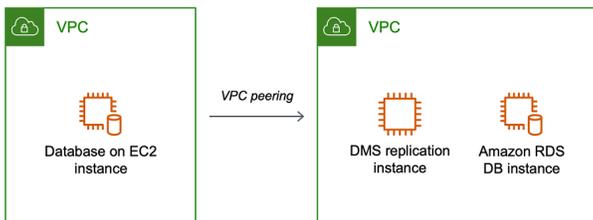
Grup keamanan VPC yang digunakan dalam konfigurasi ini harus mengizinkan ingress pada port basis data dari instans replikasi. Anda dapat melakukan ini dengan beberapa cara. Anda dapat memastikan bahwa grup keamanan yang digunakan oleh instans replikasi memiliki ingress ke titik akhir. Atau Anda dapat mengizinkan rentang VPC CIDR, NAT GW Elastic IP, atau alamat IP pribadi dari instance replikasi jika Anda menggunakannya. Tetapi kami tidak menyarankan Anda menggunakan alamat IP pribadi dari contoh replikasi, karena dapat merusak replikasi Anda jika alamat IP replikasi berubah.

Konfigurasi dengan beberapa VPC

Jika titik akhir sumber dan titik akhir target Anda berada di VPC yang berbeda, Anda dapat membuat instans replikasi di salah satu VPC tersebut. Anda kemudian dapat menghubungkan kedua VPC dengan menggunakan peering VPC.

Koneksi peering VPC adalah koneksi jaringan antara dua VPC yang memungkinkan perutean menggunakan alamat IP privat masing-masing VPC seolah-olah kedua VPC tersebut berada dalam jaringan yang sama. Anda dapat membuat koneksi peering VPC antara VPC Anda sendiri, dengan VPC di akun lain AWS, atau dengan VPC di Wilayah yang berbeda. AWS Untuk informasi lebih lanjut mengenai peering VPC, lihat [peering VPC](#) dalam Panduan Pengguna Amazon VPC.

Ilustrasi berikut menunjukkan sebuah contoh konfigurasi yang menggunakan peering VPC. Di sini, basis data sumber pada instans Amazon EC2 dalam VPC terhubung ke VPC melalui peering VPC. VPC ini berisi instans replikasi dan basis data target pada instans DB Amazon RDS.



Untuk menerapkan peering VPC, ikuti petunjuk di Bekerja [dengan koneksi peering VPC yang terletak di Amazon Virtual Private Cloud, dokumentasi VPC Peering](#). Pastikan tabel rute satu VPC berisi blok CIDR yang lain. Misalnya, jika VPC A menggunakan tujuan 10.0.0.0/16 dan VPC B menggunakan destination 172.31.0.0, tabel rute VPC A harus berisi 172.31.0.0, dan tabel rute VPC B harus berisi 10.0.0.0/16. Untuk informasi selengkapnya, lihat [Memperbarui tabel rute Anda untuk koneksi peering VPC](#) di Amazon Virtual Private Cloud, dokumentasi VPC Peering.

Grup keamanan VPC yang digunakan dalam konfigurasi ini harus mengizinkan masuknya pada port database dari instance replikasi, atau harus memungkinkan masuknya pada blok CIDR untuk VPC yang diintip.

Konfigurasi dengan VPC bersama

AWS DMS memperlakukan subnet yang dibagikan ke akun pelanggan yang berpartisipasi dalam suatu organisasi seperti subnet biasa di akun yang sama. Di bawah ini adalah deskripsi bagaimana AWS DMS menangani VPC, subnet, dan bagaimana Anda dapat menggunakan VPC bersama.

Anda dapat mengonfigurasi konfigurasi jaringan Anda untuk beroperasi di subnet khusus atau VPC dengan membuat `ReplicationSubnetGroup` objek. Saat Anda membuat `ReplicationSubnetGroup`, Anda dapat memilih untuk menentukan subnet dari VPC tertentu di akun Anda. Daftar subnet yang Anda tentukan harus menyertakan setidaknya dua subnet yang berada di zona ketersediaan terpisah, dan semua subnet harus berada dalam VPC yang sama. Saat membuat `ReplicationSubnetGroup`, pelanggan hanya menentukan subnet. AWS DMS akan menentukan VPC atas nama Anda, karena setiap subnet ditautkan ke tepat satu VPC.

Ketika Anda membuat AWS DMS `ReplicationInstance` atau a AWS DMS `ReplicationConfig`, Anda dapat memilih untuk menentukan `ReplicationSubnetGroup` dan/atau Grup Keamanan VPC di mana `ReplicationInstance` atau Replikasi Tanpa Server beroperasi. Jika tidak ditentukan, AWS DMS pilih default pelanggan `ReplicationSubnetGroup` (yang AWS DMS dibuat atas nama Anda jika tidak ditentukan untuk semua subnet dalam VPC default) dan Grup Keamanan VPC default.

Anda dapat memilih untuk menjalankan migrasi di zona ketersediaan yang Anda tentukan, atau di salah satu zona ketersediaan di zona ketersediaan. `ReplicationSubnetGroup` Saat AWS DMS mencoba membuat Instance Replikasi atau memulai Replikasi Tanpa Server, itu menerjemahkan zona ketersediaan subnet Anda ke dalam zona ketersediaan di akun layanan inti, untuk memastikan bahwa kami meluncurkan instance di Availability Zone yang benar meskipun pemetaan Availability Zone tidak identik antara kedua akun tersebut.

Jika Anda menggunakan VPC bersama, Anda harus memastikan bahwa Anda membuat `ReplicationSubnetGroup` objek yang dipetakan ke subnet yang ingin Anda gunakan dari VPC bersama. Saat membuat `ReplicationInstance` atau a `ReplicationConfig`, Anda harus menentukan `ReplicationSubnetGroup` untuk VPC bersama, dan menentukan grup keamanan VPC yang Anda buat untuk VPC bersama dengan permintaan Buat.

Perhatikan hal berikut tentang penggunaan VPC bersama:

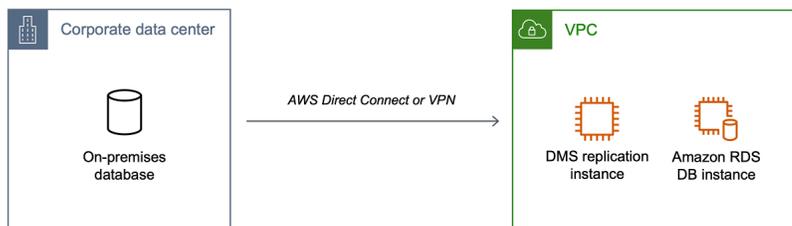
- Pemilik VPC tidak dapat berbagi sumber daya dengan peserta, tetapi peserta dapat membuat sumber daya layanan di subnet pemilik.

- Pemilik VPC tidak dapat mengakses sumber daya (seperti contoh replikasi) yang dibuat oleh peserta, karena semua sumber daya bersifat spesifik akun. Namun, selama Anda membuat instance replikasi di VPC bersama, VPC dapat mengakses sumber daya di VPC terlepas dari akun yang dimilikinya, selama titik akhir atau tugas replikasi memiliki izin yang benar.
- Karena sumber daya bersifat spesifik akun, peserta lain tidak dapat mengakses sumber daya yang dimiliki oleh akun lain. Tidak ada izin yang dapat Anda berikan kepada akun lain agar mereka dapat mengakses sumber daya yang dibuat di VPC bersama dengan akun Anda.

Konfigurasi untuk jaringan ke VPC menggunakan AWS Direct Connect atau VPN

Jaringan jarak jauh dapat terhubung ke VPC menggunakan beberapa opsi seperti Direct AWS Connect atau perangkat lunak atau koneksi VPN perangkat keras. Opsi tersebut sering digunakan untuk mengintegrasikan layanan di tempat yang sudah ada, seperti pemantauan, autentikasi, keamanan, data, atau sistem lainnya, dengan memperluas jaringan internal ke AWS cloud. Dengan menggunakan ekstensi jaringan jenis ini, Anda dapat terhubung dengan mulus ke sumber daya yang di-host AWS seperti VPC.

Ilustrasi berikut menunjukkan konfigurasi di mana titik akhir sumber adalah basis data on premise di pusat data perusahaan. Titik akhir terhubung dengan menggunakan AWS Direct Connect atau VPN ke VPC yang berisi instans replikasi dan basis data target pada instans DB Amazon RDS.

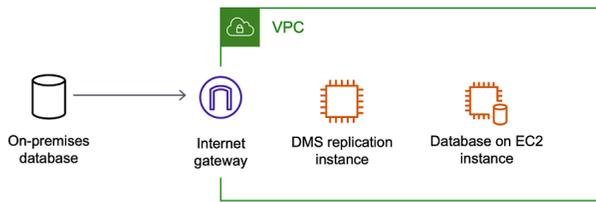


Dalam konfigurasi ini, grup keamanan VPC harus menyertakan aturan perutean yang mengirimkan lalu lintas yang ditujukan untuk rentang CIDR VPC atau alamat IP tertentu ke host. Host ini harus dapat menjembatani lalu lintas dari VPC ke VPN on premise. Dalam kasus ini, host NAT mencakup pengaturan grup keamanan sendiri. Pengaturan ini harus memungkinkan lalu lintas dari rentang CIDR VPC instance replikasi, atau alamat IP pribadi, atau grup keamanan ke dalam instance NAT. Tetapi kami tidak menyarankan Anda menggunakan alamat IP pribadi dari contoh replikasi, karena dapat merusak replikasi Anda jika alamat IP replikasi berubah.

Konfigurasi untuk jaringan ke VPC menggunakan internet

Jika Anda tidak menggunakan VPN atau terhubung AWS Direct Connect ke AWS sumber daya, Anda dapat menggunakan internet untuk memigrasi basis data Anda. Dalam hal ini, Anda dapat bermigrasi

ke instans Amazon EC2 atau instans DB Amazon RDS. Konfigurasi ini melibatkan instans replikasi publik di VPC dengan gateway internet yang memuat titik akhir target dan instans replikasi.



Untuk menambahkan gateway internet ke VPC Anda, lihat [Melampirkan gateway internet](#) di Panduan Pengguna Amazon VPC.

Tabel rute VPC harus menyertakan aturan perutean yang mengirim lalu lintas yang tidak ditujukan untuk VPC secara default ke gateway internet. Dalam konfigurasi ini, koneksi ke titik akhir tampaknya datang dari alamat IP publik instans replikasi, bukan alamat IP privat. Untuk informasi lebih lanjut, lihat [Tabel Rute VPC](#) di Panduan Pengguna Amazon VPC.

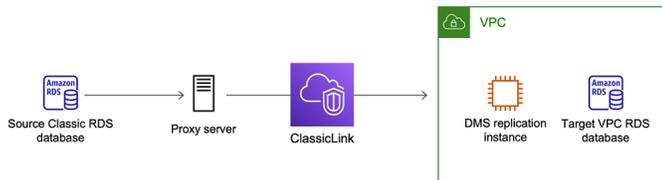
Konfigurasi dengan instans RDS DB tidak dalam VPC ke instance DB di VPC menggunakan ClassicLink

Kami pensiun EC2-Classic pada 15 Agustus 2022. Kami menyarankan Anda bermigrasi dari EC2-Classic ke VPC. Untuk mengetahui informasi selengkapnya, lihat [Migrasi dari EC2-Classic ke VPC](#) di Panduan Pengguna Amazon EC2 dan blog [Jaringan EC2-Classic akan Segera Dihentikan – Berikut Cara Mempersiapkannya](#).

Untuk menghubungkan instans Amazon RDS DB yang tidak dalam VPC ke server replikasi DMS dan instans DB di VPC, Anda dapat menggunakannya dengan server proxy. ClassicLink

ClassicLink memungkinkan Anda untuk menautkan instans EC2-Classic DB ke VPC di akun Anda, dalam Wilayah yang sama. AWS Setelah Anda membuat tautan, instans DB sumber dapat berkomunikasi dengan instans replikasi dalam VPC menggunakan alamat IP privat mereka.

Karena instance replikasi di VPC tidak dapat langsung mengakses instans DB sumber pada platform EC2-Classic ClassicLink menggunakan, Anda menggunakan server proxy. Server proksi menghubungkan instans DB sumber ke VPC yang memuat instans replikasi dan instans DB target. Server proxy menggunakan ClassicLink untuk terhubung ke VPC. Penerusan port pada server proksi memungkinkan komunikasi antara instans DB sumber dan instans DB target di VPC.



Menggunakan ClassicLink dengan AWS Database Migration Service

Anda dapat menghubungkan instans Amazon RDS DB yang tidak ada dalam VPC ke AWS server replikasi DMS dan instans DB yang ada di VPC. Untuk melakukannya, Anda dapat menggunakan Amazon EC2 ClassicLink dengan server proxy.

Prosedur berikut menunjukkan cara menggunakan ClassicLink untuk tujuan ini. Prosedur ini menghubungkan instans DB sumber Amazon RDS yang tidak ada dalam VPC ke VPC yang berisi AWS instance replikasi DMS dan instans DB target.

- Buat instance replikasi AWS DMS di VPC. (Semua instans replikasi dibuat dalam VPC.)
- Kaitkan grup keamanan VPC ke instans replikasi dan instans DB target. Ketika dua instans berbagi grup keamanan VPC, instans tersebut dapat berkomunikasi satu sama lain secara default.
- Siapkan server proksi pada instans EC2 Classic.
- Buat koneksi menggunakan ClassicLink antara server proxy dan VPC.
- Buat titik akhir AWS DMS untuk basis data sumber dan target.
- Buat tugas AWS DMS.

Untuk digunakan ClassicLink untuk memigrasikan database pada instance DB bukan di VPC ke database pada instance DB di VPC

1. Buat instance replikasi AWS DMS dan tetapkan grup keamanan VPC:

- Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.

Jika Anda masuk sebagai pengguna AWS Identity and Access Management (IAM), pastikan Anda memiliki izin yang sesuai untuk mengakses. AWS DMS Untuk informasi lebih lanjut tentang izin yang diperlukan untuk migrasi basis data, lihat [Izin IAM diperlukan untuk menggunakan AWS DMS](#).

- Pada halaman Dasbor, pilih Instans Replikasi. Ikuti petunjuk di [Langkah 1: Membuat instans replikasi menggunakan konsol AWS DMS tersebut](#) untuk membuat instans replikasi.

- c. Setelah Anda membuat instance replikasi AWS DMS, buka konsol layanan EC2. C Antarmuka Jaringan dari panel navigasi.
 - d. Pilih DMS NetworkInterface, lalu pilih Change Security Groups dari menu Actions.
 - e. Pilih grup keamanan yang ingin Anda gunakan untuk instans replikasi dan instans DB target.
2. Kaitkan grup keamanan dari langkah terakhir dengan instans DB target:
 - a. Buka konsol layanan Amazon RDS. Pilih Instans dari panel navigasi.
 - b. Pilih instans DB target. Untuk Tindakan Instans, pilih Ubah.
 - c. Untuk parameter Grup Keamanan, pilih grup keamanan yang Anda gunakan pada langkah sebelumnya.
 - d. Pilih Lanjutkan, dan kemudian Ubah Instans DB.
 3. Langkah 3: Siapkan server proksi pada instans EC2 Classic menggunakan NGINX. Gunakan AMI pilihan Anda untuk meluncurkan instans EC2 Classic. Contoh di bawah ini didasarkan pada AMI Ubuntu Server 14.04 LTS (HVM).

Untuk menyiapkan server proksi pada instans EC2 Classic

- a. Hubungkan ke instans EC2 Classic dan instal NGINX menggunakan perintah berikut:

```
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. Edit file daemon NGINX, `/etc/init/nginx.conf`, menggunakan kode berikut:

```
# /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. Buat file konfigurasi NGINX di `/usr/local/nginx/conf/nginx.conf`. Di file konfigurasi, tambahkan berikut ini:

```
# /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. Dari baris perintah, mulai NGINX menggunakan perintah berikut:

```
Prompt> sudo initctl reload-configuration
Prompt> sudo initctl list | grep nginx
Prompt> sudo initctl start nginx
```

4. Buat ClassicLink koneksi antara server proxy dan VPC target yang berisi instans DB target dan instance replikasi:
 - a. Buka konsol EC2 dan pilih instans EC2 Classic yang menjalankan server proksi.
 - b. Untuk Tindakan, pilih ClassicLink, lalu pilih Tautkan ke VPC.
 - c. Pilih grup keamanan yang Anda gunakan sebelumnya dalam prosedur ini.
 - d. Pilih Tautkan ke VPC.
5. Langkah 5: Buat titik akhir AWS DMS menggunakan prosedur di [Langkah 2: Tentukan titik akhir sumber dan target](#) Pastikan untuk menggunakan nama host DNS EC2 internal dari proksi sebagai nama server saat menentukan titik akhir sumber.
6. Buat tugas AWS DMS menggunakan prosedur di [Langkah 3: Buat tugas dan migrasi data](#).

Membuat grup subnet replikasi

Sebagai bagian dari jaringan yang akan digunakan untuk migrasi database, Anda perlu menentukan subnet mana di virtual private cloud (VPC) yang akan Anda gunakan. VPC tersebut harus berdasarkan layanan Amazon VPC. Subnet adalah serangkaian alamat IP di VPC Anda dalam Availability Zone tertentu. Subnet ini dapat didistribusikan di antara Availability Zones untuk AWS Wilayah tempat VPC Anda berada.

Saat membuat instance replikasi atau profil instans di konsol AWS DMS, Anda dapat menggunakan subnet yang Anda pilih.

Anda membuat grup subnet replikasi untuk menentukan subnet mana yang digunakan. Anda harus menentukan subnet di setidaknya dua Availability Zone.

Untuk membuat grup subnet replikasi

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.

Jika Anda masuk sebagai pengguna IAM, pastikan Anda memiliki izin yang sesuai untuk mengakses AWS DMS. Untuk informasi lebih lanjut tentang izin yang diperlukan untuk migrasi basis data, lihat [Izin IAM diperlukan untuk menggunakan AWS DMS](#).

2. Di panel navigasi, pilih Grup subnet.
3. Pilih Buat grup subnet.
4. Pada halaman grup subnet Buat replikasi, tentukan informasi grup subnet replikasi Anda. Tabel berikut menjelaskan pengaturan.

Opsi	Tindakan
Nama	Masukkan nama untuk grup subnet replikasi yang berisi antara 8 hingga 16 karakter ASCII yang dapat dicetak (kecuali /, ", dan @). Nama harus unik untuk akun Anda untuk AWS Wilayah yang Anda pilih. Anda dapat memilih untuk menambahkan beberapa kecerdasan ke nama seperti menyertakan AWS Wilayah dan tugas yang Anda lakukan, misalnya DMS-default-VPC .
Deskripsi	Masukkan deskripsi singkat dari grup subnet replikasi.
VPC	Pilih VPC yang ingin Anda gunakan untuk migrasi database. Ingatlah bahwa VPC harus memiliki minimal satu subnet di setidaknya dua Availability Zone.
Tambahkan subnet	Pilih subnet yang ingin Anda sertakan dalam grup subnet replikasi. Anda harus memilih subnet dalam setidaknya dua Availability Zone.

5. Pilih Buat grup subnet.

Menyelesaikan titik akhir domain menggunakan DNS

Biasanya, instance AWS DMS replikasi menggunakan resolver Domain Name System (DNS) dalam instans Amazon EC2 untuk menyelesaikan titik akhir domain. Jika Anda memerlukan resolusi

DNS, Anda dapat menggunakan Amazon Route 53 Resolver. Untuk informasi lebih lanjut tentang penggunaan Route 53 DNS Resolver, lihat [Memulai dengan Route 53 Resolver](#).

Untuk informasi tentang cara menggunakan server nama on premise Anda sendiri untuk menyelesaikan titik akhir tertentu menggunakan Amazon Route 53 Resolver, lihat [Menggunakan server nama on-premise Anda sendiri](#).

Mengatur kunci enkripsi untuk instans replikasi

AWS DMS mengenkripsi penyimpanan yang digunakan oleh contoh replikasi dan informasi koneksi titik akhir. Untuk mengenkripsi penyimpanan yang digunakan oleh instance replikasi, AWS DMS menggunakan AWS KMS key yang unik untuk akun Anda. AWS Anda dapat melihat dan mengelola kunci KMS ini dengan AWS Key Management Service (AWS KMS). Anda dapat menggunakan kunci KMS default di akun (`aws/dms`) atau kunci KMS yang Anda buat. Jika Anda memiliki kunci AWS KMS enkripsi yang ada, Anda juga dapat menggunakan kunci itu untuk enkripsi.

Anda dapat menentukan kunci enkripsi Anda sendiri dengan menyediakan pengenal kunci KMS untuk mengenkripsi sumber daya DMS Anda. AWS Bila Anda menentukan kunci enkripsi Anda sendiri, akun pengguna yang digunakan untuk melakukan migrasi basis data harus memiliki akses ke kunci tersebut. Untuk informasi lebih lanjut tentang membuat kunci enkripsi Anda sendiri dan memberikan akses ke kunci enkripsi kepada pengguna, lihat [Panduan Developer AWS KMS](#).

Jika Anda tidak menentukan pengenal kunci KMS, maka AWS DMS menggunakan kunci enkripsi default Anda. KMS membuat kunci enkripsi default untuk AWS DMS untuk akun Anda AWS . AWS Akun Anda memiliki kunci enkripsi default yang berbeda untuk setiap AWS Wilayah.

Untuk mengelola kunci yang digunakan untuk mengenkripsi sumber daya AWS DMS Anda, Anda gunakan. AWS KMS Anda dapat menemukan AWS KMS di AWS Management Console dengan mencari KMS pada panel navigasi.

AWS KMS menggabungkan perangkat keras dan perangkat lunak yang aman dan sangat tersedia untuk menyediakan sistem manajemen kunci yang diskalakan untuk cloud. Dengan menggunakan AWS KMS, Anda dapat membuat kunci enkripsi dan menentukan kebijakan yang mengontrol bagaimana kunci ini dapat digunakan. AWS KMS mendukung AWS CloudTrail, sehingga Anda dapat mengaudit penggunaan kunci untuk memverifikasi bahwa kunci sedang digunakan dengan tepat. AWS KMS Kunci Anda dapat digunakan dalam kombinasi dengan AWS DMS dan AWS layanan lain yang didukung. Layanan AWS yang didukung termasuk Amazon RDS, Amazon S3, Amazon Elastic Block Store (Amazon EBS), dan Amazon Redshift.

Ketika Anda telah membuat sumber daya AWS DMS Anda dengan kunci enkripsi tertentu, Anda tidak dapat mengubah kunci enkripsi untuk sumber daya tersebut. Pastikan untuk menentukan persyaratan kunci enkripsi Anda sebelum Anda membuat sumber daya AWS DMS Anda.

Membuat instans replikasi

Tugas pertama Anda dalam migrasi basis data adalah membuat instans replikasi. Instans replikasi tersebut memerlukan daya penyimpanan dan pemrosesan yang cukup untuk melakukan tugas yang Anda tetapkan dan memigrasi data dari basis data sumber Anda ke basis data target. Ukuran yang diperlukan dari instans ini bervariasi tergantung pada jumlah data yang perlu Anda migrasi dan tugas yang perlu dilakukan instans tersebut. Untuk informasi selengkapnya tentang instans replikasi, lihat [Bekerja dengan instance AWS DMS replikasi](#).

Untuk membuat instance replikasi dengan menggunakan konsol AWS

1. Pilih contoh replikasi di panel navigasi AWS DMS konsol dan kemudian pilih Buat instance replikasi.
2. Pada halaman Buat instans replikasi, tentukan informasi instans replikasi Anda. Tabel berikut menjelaskan pengaturan yang dapat Anda buat.

Opsis	Tindakan
Nama	Masukkan nama untuk instans replikasi yang berisi antara 8 hingga 16 karakter ASCII yang dapat dicetak (kecuali /, ", dan @). Nama harus unik untuk akun Anda untuk Wilayah AWS yang Anda pilih. Anda dapat memilih untuk menambahkan beberapa kecerdasan ke nama, seperti menyertakan AWS Wilayah dan tugas yang Anda lakukan, misalnya west2-mysql-instance1 .
Nama Sumber Daya Amazon Deskriptif (ARN) - Opsional	Nama ramah untuk mengganti ARN DMS default. Anda tidak dapat memodifikasinya setelah pembuatan.
Deskripsi	Masukkan deskripsi singkat dari instans replikasi.
Kelas instance	Pilih kelas instans dengan konfigurasi yang Anda butuhkan untuk migrasi Anda. Perlu diingat bahwa

Opsi	Tindakan
	instans harus memiliki daya penyimpanan, jaringan, dan pemrosesan yang cukup untuk menyelesaikan migrasi Anda dengan sukses. Untuk informasi lebih lanjut tentang cara menentukan kelas instans yang terbaik untuk migrasi Anda, lihat Bekerja dengan instance AWS DMS replikasi .
Versi mesin	Di AWS DMS konsol, Anda dapat memilih versi mesin yang didukung yang Anda inginkan. Dari AWS CLI, instance replikasi menjalankan versi non-beta terbaru dari mesin AWS DMS replikasi kecuali Anda menentukan versi mesin yang berbeda di AWS CLI
Ketersediaan Tinggi	Gunakan parameter opsional ini untuk membuat replika siaga instans replikasi Anda di Availability Zone lain untuk dukungan failover. Jika Anda bermaksud menggunakan change data capture (CDC) atau replikasi berkelanjutan, Anda harus mengaktifkan opsi ini.

Opsi	Tindakan
Penyimpanan yang dialokasikan (GiB)	<p>Penyimpanan terutama dikonsumsi oleh berkas log dan transaksi ter-cache. Untuk transaksi cache, penyimpanan hanya digunakan ketika transaksi ter-cache perlu ditulis ke disk. Oleh karena itu, AWS DMS tidak menggunakan jumlah penyimpanan yang signifikan. Beberapa pengecualian meliputi hal berikut:</p> <ul style="list-style-type: none">• Tabel sangat besar yang menghasilkan beban transaksi yang signifikan. Memuat tabel besar dapat memakan waktu, sehingga transaksi ter-cache lebih mungkin untuk ditulis ke disk selama beban tabel besar.• Tugas yang dikonfigurasi untuk menjeda sebelum memuat transaksi ter-cache. Dalam kasus ini, semua transaksi di-cache sampai beban penuh selesai untuk semua tabel. Dengan konfigurasi ini, sejumlah penyimpanan mungkin dikonsumsi oleh transaksi ter-cache.• Tugas dikonfigurasi dengan tabel yang dimuat ke Amazon Redshift. Namun, konfigurasi ini tidak menjadi masalah ketika Amazon Aurora adalah targetnya. <p>Dalam sebagian besar kasus, alokasi penyimpanan default sudah cukup. Namun, ada baiknya untuk memperhatikan metrik terkait penyimpanan. Pastikan untuk menaikkan skala penyimpanan jika Anda mengonsumsi lebih banyak dari alokasi default.</p>

Opsi	Tindakan
Jenis jaringan	DMS mendukung jenis jaringan protokol pengalaman IPv4, dan mendukung jenis jaringan protokol pengalaman IPv4 dan IPv6 dalam mode Dual-stack. Bila Anda memiliki sumber daya yang harus berkomunikasi dengan instance replikasi Anda menggunakan jenis jaringan protokol pengalaman IPv6, gunakan mode Dual-stack. Untuk informasi tentang batasan dalam mode dual-stack, lihat Batasan untuk instans DB jaringan tumpukan ganda di panduan pengguna Layanan Amazon Relational Database Service.
VPC	Pilih VPC yang ingin Anda gunakan. Jika basis data sumber atau target Anda berada di VPC, pilih VPC tersebut. Jika basis data sumber dan target Anda berada di VPC yang berbeda, pastikan bahwa keduanya berada di subnet publik dan dapat diakses publik. Kemudian pilih VPC yang akan menjadi lokasi instans replikasi. Instans replikasi harus mampu mengakses data dalam VPC sumber. Jika basis data sumber atau target tidak berada di VPC, pilih VPC yang akan menjadi lokasi instans replikasi.
Grup Subnet Replikasi	Pilih grup subnet replikasi di VPC pilihan Anda di mana instans replikasi akan dibuat. Jika basis data sumber Anda ada dalam VPC, pilih grup subnet yang berisi basis data sumber sebagai lokasi untuk instans replikasi Anda. Untuk informasi lebih lanjut tentang grup subnet replikasi, lihat Membuat grup subnet replikasi .
Dapat diakses publik	Pilih opsi ini jika Anda ingin instans replikasi dapat diakses dari internet. Default dapat diakses publik, dan setelah opsi dipilih, Anda tidak dapat memodifikasinya setelah Anda membuat instance replikasi.

3. Pilih tab Lanjutan untuk mengatur nilai pengaturan jaringan dan enkripsi jika Anda membutuhkannya. Tabel berikut menjelaskan pengaturan.

Opsi	Tindakan
Zona ketersediaan	Pilih Availability Zone di mana basis data sumber Anda berada.
Grup Keamanan VPC	Instans replikasi dibuat dalam VPC. Jika basis data sumber Anda ada dalam VPC, pilih grup keamanan VPC yang menyediakan akses ke instans DB di mana basis data berada.
Kunci KMS	Memilih kunci enkripsi yang digunakan untuk mengenkripsi penyimpanan replikasi dan informasi koneksi. Jika Anda memilih aws/dms (Default), kunci default AWS Key Management Service (AWS KMS) yang terkait dengan akun dan AWS Region Anda akan digunakan. Deskripsi dan nomor akun Anda ditampilkan, bersama dengan ARN kunci. Untuk informasi lebih lanjut tentang penggunaan kunci enkripsi, lihat Mengatur kunci enkripsi dan menentukan izin AWS KMS .

4. Tentukan pengaturan Pemeliharaan. Tabel berikut menjelaskan pengaturan. Untuk informasi lebih lanjut tentang pengaturan pemeliharaan, lihat [Bekerja dengan jendela pemeliharaan AWS DMS](#).

Opsi	Tindakan
Upgrade versi otomatis	<p>AWS DMS tidak membedakan antara versi mayor dan minor. Misalnya, memutakhirkan dari versi 3.4.x ke 3.5.x tidak dianggap sebagai peningkatan besar, jadi semua perubahan harus kompatibel ke belakang.</p> <p>Saat pemutakhiran versi Otomatis diaktifkan, DMS secara otomatis memutakhirkan versi instance</p>

Opsi	Tindakan
	<p>replikasi selama jendela pemeliharaan jika tidak digunakan lagi.</p> <p>Ketika <code>AutoMinorVersionUpgraded</code> diaktifkan, DMS menggunakan versi mesin default saat ini ketika Anda membuat instance replikasi. Misalnya, jika Anda menyetel versi Engine ke nomor versi yang lebih rendah dari versi default saat ini, DMS menggunakan versi default.</p> <p>Jika <code>AutoMinorVersionUpgraded</code> tidak diaktifkan saat Anda membuat instance replikasi, DMS menggunakan versi mesin yang ditentukan oleh parameter versi Engine.</p>
Jendela pemeliharaan	<p>Pilih rentang waktu mingguan untuk melakukan pemeliharaan sistem, dalam Waktu Universal Terkoordinasi (UTC).</p> <p>Default: Jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam per AWS Wilayah, terjadi pada hari acak dalam seminggu.</p>

5. Pilih Buat instans replikasi.

Mengubah instans replikasi

Anda dapat mengubah pengaturan untuk instans replikasi, misalnya untuk mengubah kelas instans atau untuk meningkatkan penyimpanan.

Saat Anda mengubah instans replikasi, Anda dapat segera menerapkan perubahan. Untuk segera menerapkan perubahan, pilih opsi `Langsung` terapkan perubahan pada AWS Management Console. Atau gunakan `--apply-immediately` parameter saat memanggil AWS CLI, atau atur `ApplyImmediately` parameter `true` saat menggunakan DMS API.

Jika Anda tidak memilih untuk menerapkan perubahan dengan serta-merta, perubahan akan dimasukkan ke dalam antrean perubahan yang tertunda. Selama jendela pemeliharaan berikutnya, perubahan yang tertunda di antrean akan diterapkan.

Note

Jika Anda memilih untuk menerapkan perubahan dengan segera, perubahan apa pun pada antrean modifikasi yang tertunda juga akan diterapkan. Jika salah satu dari modifikasi yang tertunda memerlukan waktu henti, memilih Langsung terapkan perubahan dapat menyebabkan waktu henti yang tidak terduga.

Untuk memodifikasi instance replikasi dengan menggunakan konsol AWS

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Di panel navigasi, pilih Instans replikasi.
3. Pilih instans replikasi yang ingin Anda ubah. Tabel berikut menjelaskan perubahan yang dapat Anda lakukan.

Opsi	Tindakan
Nama	Anda dapat mengubah nama instans replikasi. Masukkan nama untuk instans replikasi yang berisi antara 8 hingga 16 karakter ASCII yang dapat dicetak (kecuali /, ", dan @). Nama harus unik untuk akun Anda untuk Wilayah AWS yang Anda pilih. Anda dapat memilih untuk menambahkan beberapa kecerdasan ke nama, seperti menyertakan AWS Wilayah dan tugas yang Anda lakukan, misalnya west2-mysql12mysql-instance1 .
Deskripsi	Merevisi atau memasukkan deskripsi singkat dari contoh replikasi.
Kelas instance	Anda dapat mengubah kelas instans. Pilih kelas instans dengan konfigurasi yang Anda butuhkan untuk

Opsi	Tindakan
	<p>migrasi Anda. Mengubah kelas instans membuat instans replikasi mem-boot ulang. Boot ulang ini terjadi selama jendela pemeliharaan berikutnya atau dapat segera terjadi jika Anda memilih opsi Langsung terapkan perubahan.</p> <p>Untuk informasi lebih lanjut tentang cara menentukan kelas instans yang terbaik untuk migrasi Anda, lihat Bekerja dengan instance AWS DMS replikasi.</p>
Versi mesin	Anda dapat meningkatkan versi mesin yang digunakan oleh instans replikasi. Meningkatkan versi mesin replikasi menyebabkan instans replikasi untuk menutup saat sedang ditingkatkan.
Multi-AZ	Anda dapat mengubah opsi ini untuk membuat replika siaga instans replikasi Anda di Availability Zone lainnya untuk dukungan failover atau menghapus opsi ini. Jika Anda berniat untuk menggunakan change data capture (CDC), replikasi yang sedang berlangsung, Anda harus mengaktifkan opsi ini.

Opsi	Tindakan
Penyimpanan yang dialokasikan (GiB)	<p>Penyimpanan terutama dikonsumsi oleh berkas log dan transaksi ter-cache. Untuk transaksi cache, penyimpanan hanya digunakan ketika transaksi ter-cache perlu ditulis ke disk. Oleh karena itu, AWS DMS tidak menggunakan jumlah penyimpanan yang signifikan. Beberapa pengecualian meliputi hal berikut:</p> <ul style="list-style-type: none">• Tabel sangat besar yang menghasilkan beban transaksi yang signifikan. Memuat tabel besar dapat memakan waktu, sehingga transaksi ter-cache lebih mungkin untuk ditulis ke disk selama beban tabel besar.• Tugas yang dikonfigurasi untuk menjeda sebelum memuat transaksi ter-cache. Dalam kasus ini, semua transaksi di-cache sampai beban penuh selesai untuk semua tabel. Dengan konfigurasi ini, sejumlah penyimpanan mungkin dikonsumsi oleh transaksi ter-cache.• Tugas dikonfigurasi dengan tabel yang dimuat ke Amazon Redshift. Namun, konfigurasi ini tidak menjadi masalah ketika Amazon Aurora adalah targetnya. <p>Dalam sebagian besar kasus, alokasi penyimpanan default sudah cukup. Namun, ada baiknya untuk memperhatikan metrik terkait penyimpanan dan menaikkan skala penyimpanan jika Anda mengonsumsi lebih banyak dari alokasi default.</p>

Opsi	Tindakan
Jenis jaringan	DMS mendukung jenis jaringan protokol pengalaman IPv4, dan mendukung jenis jaringan protokol pengalamatan IPv4 dan IPv6 dalam mode Dual-stack. Bila Anda memiliki sumber daya yang harus berkomunikasi dengan instance replikasi Anda menggunakan jenis jaringan protokol pengalamatan IPv6, pilih mode Dual-stack. Untuk informasi tentang batasan dalam mode dual-stack, lihat Batasan untuk instans DB jaringan tumpukan ganda di panduan pengguna Layanan Amazon Relational Database Service.
Grup Keamanan VPC	Instans replikasi dibuat dalam VPC. Jika basis data sumber Anda ada dalam VPC, pilih grup keamanan VPC yang menyediakan akses ke instans DB di mana basis data berada.

Opsi	Tindakan
Upgrade versi otomatis	<p>AWS DMS tidak membedakan antara versi mayor dan minor. Misalnya, memutakhirkan dari versi 3.4.x ke 3.5.x tidak dianggap sebagai peningkatan besar, jadi semua perubahan harus kompatibel ke belakang. Saat pemutakhiran versi Otomatis diaktifkan, DMS secara otomatis memutakhirkan versi instance replikasi selama jendela pemeliharaan jika tidak digunakan lagi.</p> <p>Ketika upgrade versi otomatis diaktifkan, DMS menggunakan versi mesin default saat ini ketika Anda membuat instance replikasi. Misalnya, jika Anda menyetel versi Engine ke nomor versi yang lebih rendah dari versi default saat ini, DMS menggunakan versi default.</p> <p>Jika pemutakhiran versi Otomatis tidak diaktifkan saat Anda membuat instance replikasi, DMS menggunakan versi mesin yang ditentukan oleh parameter versi Engine.</p>
Jendela pemeliharaan	<p>Pilih rentang waktu mingguan untuk melakukan pemeliharaan sistem, dalam Waktu Universal Terkoordinasi (UTC).</p> <p>Default: Jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam per AWS Wilayah, terjadi pada hari acak dalam seminggu.</p>

Opsi	Tindakan
Terapkan perubahan segera	<p>Pilih opsi ini untuk langsung menerapkan modifikasi yang Anda buat. Tergantung pada pengaturan yang Anda pilih, memilih opsi ini dapat menyebabkan instans replikasi langsung mem-boot ulang.</p> <p>Jika Anda memilih Uji koneksi saat AWS DMS menerapkan perubahan, maka Anda akan melihat pesan kesalahan. Setelah AWS DMS menerapkan perubahan pada instance replikasi Anda, pilih Uji koneksi lagi.</p>
Terapkan perubahan selama jendela pemeliharaan terjadwal berikutnya	<p>Pilih opsi ini jika Anda ingin DMS menunggu hingga jendela pemeliharaan terjadwal berikutnya untuk menerapkan perubahan Anda.</p>

Mem-boot ulang instans replikasi

Anda dapat me-reboot instance AWS DMS replikasi untuk me-restart mesin replikasi. Boot ulang akan menyebabkan instans replikasi mati sementara, selama status instans diatur ke Bootulang. Jika AWS DMS instance dikonfigurasi untuk Multi-AZ, reboot dapat dilakukan dengan failover. AWS DMS Peristiwa dibuat saat reboot selesai.

Jika AWS DMS instans Anda adalah penerapan Multi-AZ, Anda dapat memaksa failover yang direncanakan dari satu AWS Availability Zone ke Availability Zone lainnya saat melakukan reboot. Bila Anda memaksa failover terencana dari AWS DMS instans Anda, AWS DMS menutup koneksi aktif pada instance saat ini sebelum secara otomatis beralih ke instans siaga di Availability Zone lain. Mem-boot ulang dengan failover terencana membantu Anda mensimulasikan peristiwa failover yang direncanakan dari sebuah AWS DMS instance, seperti saat menskalakan kelas instance replikasi.

Note

Setelah boot ulang memaksa failover dari satu Availability Zone ke lainnya, perubahan Availability Zone mungkin tidak tercermin selama beberapa menit. Kelambatan ini muncul di AWS Management Console, dan dalam panggilan ke AWS CLI dan AWS DMS API.

Jika tugas migrasi berjalan pada instance replikasi saat reboot terjadi, tidak ada kehilangan data yang terjadi tetapi tugas berhenti, dan status tugas berubah menjadi status kesalahan.

Jika tabel dalam tugas migrasi berada di tengah beban massal (fase beban penuh) dan belum dimulai, tabel tersebut masuk ke status kesalahan. Tetapi tabel yang lengkap pada saat itu, tetap dalam keadaan lengkap. Ketika reboot terjadi selama fase beban penuh, kami sarankan Anda melakukan salah satu langkah di bawah ini.

- Hapus tabel yang berada dalam keadaan lengkap dari tugas, dan mulai ulang tugas dengan tabel yang tersisa.
- Buat tugas baru dengan tabel dalam keadaan kesalahan, dan dengan tabel yang tertunda.

Jika tabel dalam tugas migrasi berada dalam fase replikasi yang sedang berlangsung, tugas dilanjutkan setelah boot ulang selesai.

Anda tidak dapat me-reboot instance AWS DMS replikasi Anda jika statusnya tidak dalam status Tersedia. AWS DMS Instance Anda mungkin tidak tersedia karena beberapa alasan, seperti modifikasi yang diminta sebelumnya atau tindakan jendela pemeliharaan. Waktu yang diperlukan untuk me-reboot instance AWS DMS replikasi biasanya kecil (di bawah 5 menit).

Mem-boot ulang instance replikasi menggunakan konsol AWS

Untuk me-reboot instance replikasi, gunakan AWS konsol.

Untuk me-reboot instance replikasi menggunakan konsol AWS

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Di panel navigasi, pilih Instans replikasi.
3. Pilih instans replikasi yang ingin Anda boot ulang.
4. Pilih Boot ulang. Kotak dialog contoh replikasi Reboot terbuka.
5. Pilih kotak centang untuk Reboot dengan failover yang direncanakan? jika Anda telah mengonfigurasi instance replikasi untuk penerapan Multi-AZ dan Anda ingin gagal ke Availability Zone lain AWS .
6. Pilih Boot ulang.

Mem-boot ulang instans replikasi menggunakan CLI

Untuk me-reboot instance replikasi, gunakan AWS CLI [reboot-replication-instance](#) perintah dengan parameter berikut:

- `--replication-instance-arn`

Example Contoh boot ulang sederhana

AWS CLI Contoh berikut me-reboot instance replikasi.

```
aws dms reboot-replication-instance \  
--replication-instance-arn arn of my rep instance
```

Example Contoh boot ulang sederhana dengan failover

AWS CLI Contoh berikut me-reboot instance replikasi dengan failover.

```
aws dms reboot-replication-instance \  
--replication-instance-arn arn of my rep instance \  
--force-planned-failover
```

Mem-boot ulang instans replikasi menggunakan API

Untuk me-reboot instance replikasi, gunakan [RebootReplicationInstance](#) aksi AWS DMS API dengan parameter berikut:

- `ReplicationInstanceArn = arn of my rep instance`

Example Contoh boot ulang sederhana

Contoh kode berikut mem-boot ulang instans replikasi.

```
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 Contoh boot ulang sederhana dengan failover

Contoh kode berikut me-reboot instance replikasi dan gagal ke AWS Availability Zone lain.

```
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
```

Menghapus instans replikasi

Anda dapat menghapus contoh AWS DMS replikasi ketika Anda selesai menggunakannya. Jika Anda memiliki tugas migrasi yang menggunakan instans replikasi, Anda harus menghentikan dan menghapus tugas-tugas tersebut sebelum menghapus instans replikasi.

Jika Anda menutup AWS akun, semua AWS DMS sumber daya dan konfigurasi yang terkait dengan akun Anda akan dihapus setelah dua hari. Sumber daya tersebut mencakup semua instans replikasi, konfigurasi titik akhir sumber dan target, tugas replikasi, dan sertifikat SSL. Jika setelah dua hari Anda memutuskan untuk menggunakan AWS DMS lagi, Anda membuat ulang sumber daya yang Anda butuhkan.

Jika instance replikasi Anda memenuhi semua kriteria untuk penghapusan, dan tetap dalam DELETING status untuk jangka waktu yang lama, hubungi dukungan untuk memecahkan masalah.

Menghapus instance replikasi menggunakan konsol AWS

Untuk menghapus instance replikasi, gunakan AWS konsol.

Untuk menghapus instance replikasi menggunakan konsol AWS

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Di panel navigasi, pilih Instans replikasi.
3. Pilih instans replikasi yang ingin dihapus.
4. Pilih Hapus.
5. Di kotak dialog, pilih Hapus.

Menghapus instans replikasi menggunakan CLI

Untuk menghapus contoh replikasi, gunakan AWS CLI [delete-replication-instance](#) perintah dengan parameter berikut:

- `--replication-instance-arn`

Example Contoh hapus

AWS CLI Contoh berikut menghapus contoh replikasi.

```
aws dms delete-replication-instance \  
--replication-instance-arn arn of my rep instance
```

Menghapus instans replikasi menggunakan API

Untuk menghapus instance replikasi, gunakan [DeleteReplicationInstance](#) aksi AWS DMS API dengan parameter berikut:

- `ReplicationInstanceArn = arn of my rep instance`

Example Contoh hapus

Contoh kode berikut menghapus instans replikasi.

```
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
```

Bekerja dengan jendela pemeliharaan AWS DMS

Setiap contoh AWS DMS replikasi memiliki jendela pemeliharaan mingguan di mana setiap perubahan sistem yang tersedia diterapkan. Anda dapat menganggap jendela pemeliharaan sebagai peluang untuk mengontrol ketika modifikasi dan patching perangkat lunak terjadi.

Jika AWS DMS menentukan bahwa pemeliharaan diperlukan selama minggu tertentu, pemeliharaan terjadi selama jendela pemeliharaan 30 menit yang Anda pilih saat Anda membuat instance replikasi. AWS DMS menyelesaikan sebagian besar perawatan selama jendela pemeliharaan 30 menit. Namun, waktu yang lebih lama mungkin diperlukan untuk perubahan yang lebih besar.

Pengaruh pemeliharaan pada tugas migrasi yang ada

Saat tugas AWS DMS migrasi berjalan pada sebuah instance, peristiwa berikut terjadi saat tambalan diterapkan:

- Jika tabel dalam tugas migrasi berada dalam fase mereplikasi perubahan berkelanjutan (CDC), AWS DMS hentikan tugas sejenak dan kemudian lanjutkan setelah tambalan diterapkan. Migrasi kemudian berlanjut dari tempat di mana migrasi terganggu ketika patch diterapkan.
- Jika AWS DMS memigrasikan tabel sebagai bagian dari migrasi data yang ada atau memigrasikan data yang ada dan mereplikasi tugas perubahan yang sedang berlangsung, DMS berhenti dan kemudian memulai ulang migrasi untuk semua tabel yang berada dalam fase pemuatan penuh saat tambalan diterapkan. DMS juga berhenti dan melanjutkan semua tabel yang berada dalam fase CDC saat patch diterapkan.

Mengubah pengaturan jendela pemeliharaan

Anda dapat mengubah kerangka waktu jendela pemeliharaan menggunakan AWS Management Console, AWS CLI, atau AWS DMS API.

Mengubah pengaturan jendela pemeliharaan menggunakan konsol

Anda dapat mengubah jangka waktu jendela pemeliharaan menggunakan AWS Management Console.

Untuk mengubah jendela pemeliharaan yang diinginkan menggunakan konsol

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Di panel navigasi, pilih Instans replikasi.
3. Pilih instans replikasi yang ingin Anda ubah dan pilih Modifikasi.
4. Perluas tab Pemeliharaan dan pilih tanggal dan waktu untuk jendela pemeliharaan Anda.
5. Pilih Langsung terapkan perubahan.
6. Pilih Modifikasi.

Mengubah pengaturan jendela pemeliharaan menggunakan CLI

Untuk menyesuaikan jendela pemeliharaan yang disukai, gunakan AWS CLI [modify-replication-instance](#) perintah dengan parameter berikut.

- `--replication-instance-identifier`
- `--preferred-maintenance-window`

Example

AWS CLI Contoh berikut menetapkan jendela pemeliharaan ke hari Selasa dari pukul 4:00 - 4:30 pagi. UTC.

```
aws dms modify-replication-instance \  
--replication-instance-identifier myreinstance \  
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

Mengubah pengaturan jendela pemeliharaan menggunakan API

Untuk menyesuaikan jendela pemeliharaan yang diinginkan, gunakan [ModifyReplicationInstance](#) tindakan AWS DMS API dengan parameter berikut.

- `ReplicationInstanceIdentifier` = *myreinstance*

- PreferredMaintenanceWindow = *Tue:04:00-Tue:04:30*

Example

Contoh kode berikut mengatur jendela pemeliharaan ke Selasa mulai 4:00-4:30 pagi. 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
```

Bekerja dengan Titik akhir DMS AWS

Titik akhir menyediakan koneksi, tipe menyimpan data, dan informasi lokasi untuk menyimpan data Anda. AWS Database Migration Service menggunakan informasi ini untuk sebagai connect ke tempat menyimpan data dan melakukan migrasi data dari titik akhir sumber ke titik akhir target. Anda dapat menentukan atribut koneksi tambahan untuk titik akhir dengan menggunakan pengaturan titik akhir. Pengaturan ini dapat mengontrol pencatatan, ukuran file, dan parameter lainnya; untuk informasi selengkapnya tentang pengaturan titik akhir, lihat bagian dokumentasi untuk penyimpanan data Anda.

Berikut ini, Anda dapat mengetahui detail lebih lanjut tentang titik akhir.

Topik

- [Membuat titik akhir sumber dan target](#)
- [Sumber untuk migrasi data](#)
- [Target migrasi data](#)
- [Mengkonfigurasi titik akhir VPC sebagaiAWSSumber DMS dan titik akhir target](#)
- [Pernyataan DDL didukung oleh AWS DMS](#)

Membuat titik akhir sumber dan target

Anda dapat membuat titik akhir sumber dan target ketika membuat instans replikasi atau Anda dapat membuat titik akhir setelah instans replikasi dibuat. Penyimpanan sumber dan data target dapat dilakukan pada instans Amazon Elastic Compute Cloud (Amazon EC2), instans DB Amazon Relational Database Service (Amazon RDS), atau basis data on premise. (Perhatikan bahwa salah satu titik akhir harus berada di layanan AWS. Anda tidak dapat menggunakan DMS AWS untuk melakukan migrasi dari basis data on premise ke basis data on premise lainnya.)

Prosedur berikut mengasumsikan bahwa Anda telah memilih wizard konsol DMS AWS. Perhatikan bahwa Anda juga dapat melakukan langkah ini dengan memilih Titik akhir dari Panel navigasi konsol DMS AWS dan kemudian memilih Membuat titik akhir. Saat menggunakan wizard konsol tersebut, Anda membuat titik akhir sumber dan target pada halaman yang sama. Bila tidak menggunakan wizard konsol tersebut, Anda membuat setiap titik akhir secara terpisah.

Untuk menentukan titik akhir basis data sumber atau target menggunakan konsol AWS tersebut

1. Pada Connect pada titik akhir basis data sumber dan target, tentukan informasi koneksi Anda untuk basis data sumber atau target. Tabel berikut menjelaskan pengaturan.

Untuk opsi ini	Lakukan ini
Jenis titik akhir	Memilih apakah titik akhir ini adalah titik akhir sumber atau target.
Pilih Instans RDS DB	Memilih opsi ini jika titik akhir adalah instans DB Amazon RDS.
Pengidentifikasi titik akhir	Ketik nama yang ingin Anda gunakan untuk mengidentifikasi titik akhir. Anda mungkin ingin memasukkan tipe titik akhir dalam nama tersebut, seperti oracle-source atau PostgreSQL-target . Nama itu harus unik untuk semua instans replikasi.
Mesin sumber dan Mesin target	Memilih tipe mesin basis data yaitu titik akhir.
Akses ke database endpoint	<p>Memilih opsi yang ingin Anda gunakan untuk menentukan kredensial basis data titik akhir:</p> <ul style="list-style-type: none"> • Memilih AWS Secrets Manager— Gunakan beberapa rahasia yang ditentukan dalam AWS Secrets Manager untuk secara rahasia memberikan kredensial Anda seperti yang ditunjukkan berikut ini. Untuk informasi lebih lanjut tentang menciptakan rahasia ini dan peran mengakses rahasia yang mengaktifkan AWS DMS untuk mengakses rahasia tersebut, lihat Menggunakan rahasia untuk mengakses titik akhir AWS Database Migration Service. • Berikan informasi untuk mengakses secara manual— Menggunakan kredensial bebas teks agar Anda memasukkan langsung seperti yang ditunjukkan berikut.

Untuk opsi ini	Lakukan ini
Memilih AWS Secrets Manager	Mengatur kredensial rahasia berikut.
ID Rahasia	Ketik Amazon Resource Name (ARN) lengkap, ARN parsial, atau nama yang dikenal dari suatu rahasia yang telah Anda buat dalam AWS Secrets Manager untuk mengakses basis data titik akhir.
Peran IAM	Ketik ARN dari peran mengakses rahasia yang telah Anda buat dalam IAM untuk mengakses AWS DMS atas nama Anda yang diidentifikasi oleh ID Rahasia. Untuk informasi tentang membuat peran akses rahasia, lihat Menggunakan rahasia untuk mengakses titik akhir AWS Database Migration Service .
ID Rahasia untuk manajemen penyimpanan otomatis Oracle (ASM)	(Untuk titik akhir sumber Oracle yang menggunakan Oracle ASM saja) Ketik Amazon Resource Name (ARN) lengkap, ARN sebagian, atau nama rahasia yang dikenal yang telah Anda buat di AWS Secrets Manager untuk mengakses Oracle ASM. Rahasia ini biasanya dibuat untuk mengakses Oracle ASM pada server yang sama sebagaimana rahasia diidentifikasi oleh ID rahasia.
Peran IAM untuk Oracle ASM	(Untuk titik akhir sumber Oracle menggunakan Oracle ASM saja) Ketik ARN dari peran mengakses rahasia yang telah Anda buat di IAM untuk mengakses AWS DMS atas nama Anda ke rahasia yang diidentifikasi oleh ID rahasia untuk mengelola penyimpanan otomatis Oracle (ASM).
Memberikan informasi untuk mengakses secara manual	Mengatur kredensial bebas teks berikut.

Untuk opsi ini	Lakukan ini
Nama server	Ketik nama server. Untuk basis data on premise, ini bisa berupa alamat IP atau nama host publik. Untuk instans DB Amazon RDS, ini bisa berupa titik akhir (juga disebut nama DNS) untuk instans DB, seperti mysqlsrvinst.abcd12345678.us-west-2.rds.amazonaws.com .
Port	Ketik port yang digunakan oleh basis data.
Mode Secure Socket Layer (SSL)	Memilih mode SSL jika Anda ingin mengaktifkan enkripsi koneksi untuk titik akhir ini. Tergantung pada mode pilihan Anda, Anda mungkin diminta untuk memberikan informasi sertifikat dan server sertifikat.
Nama pengguna	Ketik dengan mengizinkan nama pengguna yang diperlukan untuk mengizinkan migrasi data. Untuk informasi tentang izin yang diperlukan, lihat bagian keamanan untuk mesin basis data sumber atau target dalam panduan pengguna ini.
Kata Sandi	Ketik kata sandi untuk akun dengan izin yang diperlukan. Kata sandi untuk titik akhir sumber dan target AWS DMS memiliki batasan karakter, tergantung pada mesin basis data. Untuk informasi lebih lanjut, lihat tabel di bawah ini.
Nama basis data	Untuk mesin basis data tertentu, nama basis data yang ingin Anda gunakan sebagai titik akhir basis data.

Tabel berikut mencantumkan karakter yang tidak didukung dalam kata sandi endpoint dan rahasia manajer rahasia untuk mesin database yang terdaftar. Jika Anda ingin menggunakan koma (,) dalam kata sandi titik akhir, gunakan support Secrets Manager yang disediakan di AWS DMS untuk autentikasi dalam mengakses ke instans AWS DMS Anda. Untuk informasi

selengkapnya, lihat [Menggunakan rahasia untuk mengakses titik akhir AWS Database Migration Service](#).

Untuk mesin basis data ini	Karakter berikut tidak didukung dalam kata sandi titik akhir dan rahasia manajer rahasia
Semua	{ }
Microsoft Azure, hanya sebagai sumber	;
Microsoft SQL Server	, ;
Kompatibel MySQL, termasuk MySQL, MariaDB, dan Amazon Aurora MySQL	;
Oracle	,
PostgreSQL, Amazon Aurora PostgreSQL Edisi yang kompatibel, dan Amazon Aurora Tanpa Server sebagai target hanya untuk Edisi yang kompatibel dengan Aurora PostgreSQL	; + %
Amazon Redshift Sebagai target saja	, ;

- Pilih pengaturan Endpoint dan AWS KMS key jika Anda membutuhkannya. Anda dapat menguji koneksi titik akhir dengan memilih Jalankan pengujian. Tabel berikut menjelaskan pengaturan.

Untuk opsi ini	Lakukan hal berikut
Pengaturan titik akhir	Pilih parameter koneksi tambahan di sini. Untuk informasi selengkapnya tentang pengaturan titik akhir, lihat bagian dokumentasi untuk mesin Sumber atau mesin Target Anda (ditentukan pada langkah 1).

Untuk opsi ini	Lakukan hal berikut
	<p>Untuk endpoint sumber Oracle yang menggunakan Oracle ASM, jika Anda memilih Menyediakan informasi akses secara manual di langkah 1, Anda mungkin juga perlu mengetik pengaturan endpoint untuk menentukan kredensi pengguna Oracle ASM. Untuk informasi selengkapnya tentang pengaturan endpoint Oracle ASM ini, lihat. Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC</p>
AWS KMS key	<p>Memilih kunci enkripsi yang digunakan untuk mengenkripsi penyimpanan replikasi dan informasi koneksi. Jika Anda memilih (Default) aws/dms, kunci AWS default (AWS KMS) yang terkait dengan akun Anda dan Wilayah AWS akan digunakan. Untuk informasi lebih lanjut tentang penggunaan kunci enkripsi, lihat Mengatur kunci enkripsi dan menentukan izin AWS KMS.</p>
Uji koneksi titik akhir (opsional)	<p>Menambahkan VPC dan nama instans replikasi. Untuk menguji koneksi, memilih Jalankan pengujian.</p>

Sumber untuk migrasi data

AWS Database Migration Service (AWS DMS) dapat menggunakan banyak mesin data yang paling populer sebagai sumber untuk replikasi data. Sumber basis data dapat menjadi mesin yang dikelola sendiri yang berjalan pada instans Amazon EC2 atau basis data lokal. Atau dapat berupa sumber data di layanan AWS seperti Amazon RDS atau Amazon S3.

Untuk daftar lengkap sumber yang valid, lihat [Sumber untuk AWS DMS](#).

Topik

- [Menggunakan basis data Oracle sebagai sumber untuk AWS DMS](#)
- [Menggunakan database Microsoft SQL Server sebagai sumber AWS DMS](#)
- [Menggunakan basis data Microsoft Azure SQL sebagai sumber untuk AWS DMS](#)

- [Menggunakan Microsoft Azure SQL Managed Instance sebagai sumber untuk AWS DMS](#)
- [Menggunakan Microsoft Azure Database untuk PostgreSQL server fleksibel sebagai sumber untuk AWS DMS](#)
- [Menggunakan Microsoft Azure Database untuk MySQL server fleksibel sebagai sumber AWS DMS](#)
- [Menggunakan OCI MySQL Heatwave sebagai sumber untuk AWS DMS](#)
- [Menggunakan Google Cloud untuk MySQL sebagai sumber untuk AWS DMS](#)
- [Menggunakan Google Cloud untuk PostgreSQL sebagai sumber untuk AWS DMS](#)
- [Menggunakan basis data PostgreSQL sebagai sumber AWS DMS](#)
- [Menggunakan basis data yang kompatibel dengan MySQL sebagai sumber untuk AWS DMS](#)
- [Menggunakan basis data SAP ASE sebagai sumber untuk AWS DMS](#)
- [Menggunakan MongoDB sebagai sumber untuk AWS DMS](#)
- [Menggunakan Amazon DocumentDB \(dengan kompatibilitas MongoDB\) sebagai sumber untuk AWS DMS](#)
- [Menggunakan Amazon S3 sebagai sumber untuk AWS DMS](#)
- [Menggunakan IBM Db2 untuk Linux, Unix, Windows, dan database Amazon RDS \(Db2 LUW\) sebagai sumber untuk AWS DMS](#)
- [Menggunakan IBM Db2 untuk database z/OS sebagai sumber untuk AWS DMS](#)

Menggunakan basis data Oracle sebagai sumber untuk AWS DMS

Anda dapat memigrasikan data dari satu atau banyak database Oracle menggunakan AWS DMS. Dengan basis data Oracle sebagai sumber, Anda dapat memigrasi data ke salah satu target yang didukung oleh AWS DMS.

AWS DMS mendukung edisi database Oracle berikut:

- Oracle Enterprise Edition
- Oracle Standard Edition
- Oracle Express Edition
- Oracle Personal Edition

Untuk informasi tentang versi database Oracle yang AWS DMS mendukung sebagai sumber, lihat [Sumber untuk AWS DMS](#)

Anda dapat menggunakan Lapisan Soket Aman (SSL) untuk mengenkripsi koneksi antara titik akhir Oracle Anda dan instans replikasi Anda. Untuk informasi lebih lanjut tentang menggunakan SSL dengan endpoint Oracle, lihat [Dukungan SSL untuk titik akhir Oracle](#)

AWS DMS mendukung penggunaan enkripsi data transparan Oracle (TDE) untuk mengenkripsi data saat istirahat di database sumber. Untuk informasi lebih lanjut tentang menggunakan Oracle TDE dengan titik akhir sumber Oracle, lihat [Metode enkripsi yang didukung untuk menggunakan Oracle sebagai sumber AWS DMS](#).

AWS mendukung penggunaan TLS versi 1.2 dan yang lebih baru dengan endpoint Oracle (dan semua jenis endpoint lainnya), dan merekomendasikan penggunaan TLS versi 1.3 atau yang lebih baru.

Ikuti langkah-langkah ini untuk mengonfigurasi database Oracle sebagai titik akhir AWS DMS sumber:

1. Buat pengguna Oracle dengan izin yang sesuai AWS DMS untuk mengakses database sumber Oracle Anda.
2. Buat titik akhir sumber Oracle yang sesuai dengan konfigurasi basis data Oracle yang Anda pilih. Untuk membuat full-load-only tugas, tidak diperlukan konfigurasi lebih lanjut.
3. Untuk membuat tugas yang menangani pengambilan data perubahan (tugas CDC saja atau beban penuh dan CDC), pilih Oracle LogMiner atau AWS DMS Binary Reader untuk menangkap perubahan data. Memilih LogMiner atau Pembaca Biner menentukan beberapa izin dan opsi konfigurasi selanjutnya. Untuk perbandingan LogMiner dan Pembaca Biner, lihat bagian berikut.

Note

Untuk informasi selengkapnya tentang tugas beban penuh, tugas CDC saja, dan tugas beban penuh dan CDC, lihat [Membuat tugas](#)

Untuk detail tambahan tentang bekerja dengan database sumber Oracle dan AWS DMS, lihat bagian berikut.

Topik

- [Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC](#)
- [Alur kerja untuk mengonfigurasi database sumber Oracle yang dikelola sendiri atau AWS dikelola untuk AWS DMS Mengkonfigurasi database sumber Oracle](#)

- [Bekerja dengan database Oracle yang dikelola sendiri sebagai sumber AWS DMS](#)
- [Bekerja dengan database Oracle yang AWS dikelola sebagai sumber AWS DMS](#)
- [Keterbatasan dalam menggunakan Oracle sebagai sumber AWS DMS](#)
- [Dukungan SSL untuk titik akhir Oracle](#)
- [Metode enkripsi yang didukung untuk menggunakan Oracle sebagai sumber AWS DMS](#)
- [Metode kompresi yang didukung untuk menggunakan Oracle sebagai sumber AWS DMS](#)
- [Mereplikasi tabel bersarang menggunakan Oracle sebagai sumber AWS DMS](#)
- [Menyimpan REDO di Oracle ASM saat menggunakan Oracle sebagai sumber AWS DMS](#)
- [Pengaturan titik akhir saat menggunakan Oracle sebagai sumber AWS DMS](#)
- [Jenis data sumber untuk Oracle](#)

Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC

Dalam AWS DMS, ada dua metode untuk membaca redo log ketika melakukan change data capture (CDC) untuk Oracle sebagai sumber: Oracle LogMiner dan Binary Reader. AWS DMS LogMiner adalah API Oracle untuk membaca log redo online dan file log redo yang diarsipkan. Binary Reader adalah AWS DMS metode yang membaca dan mem-parsing file log redo mentah secara langsung. Metode ini memiliki beberapa fitur berikut.

Fitur	LogMiner	Binary Reader
Mudah dikonfigurasi	Ya	Tidak
Dampak yang lebih rendah pada sistem sumber I/O dan CPU	Tidak	Ya
Performa CDC yang lebih baik	Tidak	Ya
Mendukung klaster tabel Oracle	Ya	Tidak
Mendukung semua jenis Hybrid Columnar Compression (HCC) Oracle	Ya	Sebagian Binary Reader tidak mendukung QUERY LOW untuk tugas-

Fitur	LogMiner	Binary Reader
		tugas dengan CDC. Semua jenis HCC lainnya didukung sepenuhnya.
Dukungan kolom LOB hanya di Oracle 12c	Tidak (LOB Support tidak tersedia dengan LogMiner di Oracle 12c.)	Ya
Mendukung pernyataan UPDATE yang hanya memengaruhi kolom LOB	Tidak	Ya
Mendukung enkripsi data transparan (TDE) Oracle	Sebagian Saat menggunakan Oracle LogMiner, AWS DMS tidak mendukung enkripsi TDE pada tingkat kolom untuk Amazon RDS for Oracle.	Sebagian Binary Reader mendukung TDE hanya untuk basis data Oracle yang dikelola sendiri.
Mendukung semua metode kompresi Oracle	Ya	Tidak
Mendukung transaksi XA	Tidak	Ya
RAC	Ya Tidak direkomendasikan	Ya Sangat direkomendasikan

Note

Secara default, AWS DMS menggunakan Oracle LogMiner for (CDC).

AWS DMS mendukung metode enkripsi data transparan (TDE) saat bekerja dengan database sumber Oracle. Jika kredensial TDE yang Anda tentukan salah, tugas AWS DMS migrasi tidak akan gagal, yang dapat memengaruhi replikasi tabel terenkripsi yang sedang berlangsung. Untuk informasi selengkapnya tentang menentukan kredensial TDE, lihat.

[Metode enkripsi yang didukung untuk menggunakan Oracle sebagai sumber AWS DMS](#)

Keuntungan utama menggunakan LogMiner dengan AWS DMS meliputi:

- LogMiner mendukung sebagian besar opsi Oracle, seperti opsi enkripsi dan opsi kompresi. Binary Reader tidak mendukung semua pilihan Oracle, terutama kompresi dan sebagian besar pilihan untuk enkripsi.
- LogMiner menawarkan konfigurasi yang lebih sederhana, terutama dibandingkan dengan pengaturan akses langsung Binary Reader atau ketika log redo dikelola menggunakan Oracle Automatic Storage Management (ASM).
- LogMiner mendukung kluster tabel untuk digunakan oleh AWS DMS. Binary Reader tidak mendukung kluster tabel.

Keuntungan utama menggunakan Binary Reader dengan AWS DMS meliputi:

- Untuk migrasi dengan volume perubahan yang tinggi, LogMiner mungkin memiliki beberapa dampak I/O atau CPU pada komputer yang menghosting database sumber Oracle. Binary Reader memiliki kemungkinan yang lebih sedikit untuk memiliki dampak I/O atau CPU karena log ditambang langsung daripada membuat beberapa kueri basis data.
- Untuk migrasi dengan volume perubahan yang tinggi, kinerja CDC biasanya jauh lebih baik ketika menggunakan Binary Reader dibandingkan dengan menggunakan Oracle. LogMiner
- Binary Reader mendukung CDC untuk LOB di Oracle versi 12c. LogMiner tidak.

Secara umum, gunakan Oracle LogMiner untuk memigrasikan database Oracle Anda kecuali Anda memiliki salah satu situasi berikut:

- Anda perlu menjalankan beberapa tugas migrasi pada sumber basis data Oracle.

- Volume perubahan atau volume log redo pada basis data sumber Oracle tinggi, atau Anda memiliki perubahan dan juga menggunakan ASM Oracle.

Note

Jika Anda mengubah antara menggunakan Oracle LogMiner dan AWS DMS Binary Reader, pastikan untuk memulai ulang tugas CDC.

Konfigurasi untuk CDC pada sumber basis data Oracle

Agar titik akhir sumber Oracle terhubung ke basis data untuk tugas change data capture (CDC), Anda mungkin perlu menentukan atribut koneksi tambahan. Hal yang sama juga berlaku untuk tugas beban penuh dan CDC atau untuk tugas CDC-only. Atribut koneksi tambahan yang Anda tentukan bergantung pada metode yang Anda gunakan untuk mengakses log redo: Oracle LogMiner atau AWS DMS Binary Reader.

Anda menentukan atribut sambungan tambahan saat Anda membuat titik akhir sumber. Jika Anda memiliki beberapa sambungan pengaturan atribut, pisahkan satu sama lain dengan titik koma tanpa spasi putih tambahan (misalnya, `oneSetting;thenAnother`).

AWS DMS menggunakan secara LogMiner default. Anda tidak harus menentukan atribut sambungan tambahan untuk menggunakannya.

Untuk menggunakan Binary Reader untuk mengakses log redo, tambahkan atribut sambungan tambahan berikut.

```
useLogMinerReader=N;useBfile=Y;
```

Gunakan format berikut untuk atribut sambungan tambahan untuk mengakses server yang menggunakan ASM dengan Binary Reader.

```
useLogMinerReader=N;useBfile=Y;asm_user=asm_username;asm_server=RAC_server_ip_address:port_number+ASM;
```

Atur parameter permintaan Password titik akhir sumber untuk kata sandi pengguna Oracle dan kata sandi ASM, yang dipisah dengan koma sebagai berikut.

```
oracle_user_password,asm_user_password
```

Apabila sumber Oracle menggunakan ASM, Anda dapat bekerja dengan pilihan performa tinggi pada Binary Reader untuk pemrosesan transaksi dalam skala besar. Pilihan ini mencakup atribut sambungan tambahan untuk menentukan jumlah thread paralel (`parallelASMReadThreads`) dan jumlah buffer read-ahead (`readAheadBlocks`). Pengaturan atribut ini secara bersamaan dapat secara signifikan meningkatkan performa tugas CDC. Pengaturan berikut memberikan hasil yang baik untuk sebagian besar konfigurasi ASM.

```
useLogMinerReader=N;useBfile=Y;asm_user=asm_username;asm_server=RAC_server_ip_address:port_number  
+ASM;  
parallelASMReadThreads=6;readAheadBlocks=150000;
```

Untuk informasi lebih lanjut tentang nilai-nilai yang didukung atribut sambungan tambahan, lihat [Pengaturan titik akhir saat menggunakan Oracle sebagai sumber AWS DMS](#).

Selain itu, performa tugas CDC dengan sumber Oracle yang menggunakan ASM tergantung pada pengaturan lain yang Anda pilih. Pengaturan ini mencakup atribut sambungan tambahan AWS DMS dan pengaturan SQL untuk mengonfigurasi sumber Oracle. Untuk informasi lebih lanjut tentang atribut sambungan tambahan untuk sumber Oracle yang menggunakan ASM, lihat [Pengaturan titik akhir saat menggunakan Oracle sebagai sumber AWS DMS](#)

Anda juga perlu memilih titik awal CDC yang sesuai. Biasanya ketika Anda melakukan hal ini, Anda ingin mengidentifikasi titik pemrosesan transaksi yang menangkap transaksi terbuka paling awal untuk memulai CDC. Jika tidak, tugas CDC dapat melewatkan transaksi terbuka sebelumnya. Untuk basis data sumber Oracle, Anda dapat memilih titik awal CDC berdasarkan system change number (SCN) Oracle untuk mengidentifikasi transaksi terbuka paling awal ini. Untuk informasi selengkapnya, lihat [Melakukan replikasi mulai dari titik awal CDC](#).

Untuk informasi lebih lanjut tentang mengonfigurasi CDC untuk basis data Oracle yang dikelola sendiri sebagai sumber, lihat [Hak istimewa akun diperlukan saat menggunakan Oracle LogMiner untuk mengakses log redo](#), [Hak istimewa akun diperlukan saat menggunakan AWS DMS Binary Reader untuk mengakses redo log](#), dan [Hak istimewa akun tambahan yang diperlukan saat menggunakan Binary Reader dengan Oracle ASM](#).

Untuk informasi selengkapnya tentang mengonfigurasi CDC untuk database Oracle yang AWS dikelola sebagai sumber, lihat dan [Mengkonfigurasi tugas CDC untuk menggunakan Binary Reader](#)

[dengan RDS untuk sumber Oracle AWS DMS Menggunakan Amazon RDS Oracle Standby \(replika baca\) sebagai sumber dengan Binary Reader untuk CDC di AWS DMS](#)

Alur kerja untuk mengonfigurasi database sumber Oracle yang dikelola sendiri atau AWS dikelola untuk AWS DMS

Alur kerja untuk mengonfigurasi database sumber Oracle yang dikelola sendiri atau AWS dikelola untuk AWS DMS

Untuk mengonfigurasi instans basis data sumber yang dikelola sendiri, gunakan langkah-langkah alur kerja berikut, bergantung pada bagaimana Anda melakukan CDC.

Untuk langkah alur kerja ini	Jika Anda menggunakan CDC LogMiner, lakukan ini	Jika Anda melakukan CDC menggunakan Binary Reader, lakukan ini
Berikan hak istimewa akun Oracle.	Lihat Hak istimewa akun pengguna diperlukan pada sumber Oracle yang dikelola sendiri untuk AWS DMS .	Lihat Hak istimewa akun pengguna diperlukan pada sumber Oracle yang dikelola sendiri untuk AWS DMS .
Siapkan basis data sumber untuk replikasi menggunakan CDC.	Lihat Mempersiapkan database sumber yang dikelola sendiri Oracle untuk CDC menggunakan AWS DMS .	Lihat Mempersiapkan database sumber yang dikelola sendiri Oracle untuk CDC menggunakan AWS DMS .
Berikan hak istimewa pengguna Oracle tambahan yang diperlukan untuk CDC.	Lihat Hak istimewa akun diperlukan saat menggunakan Oracle LogMiner untuk mengakses log redo .	Lihat Hak istimewa akun diperlukan saat menggunakan AWS DMS Binary Reader untuk mengakses redo log .
Untuk instans Oracle dengan ASM, berikan hak istimewa pengguna akun tambahan yang diperlukan untuk mengakses ASM untuk CDC.	Tidak ada tindakan tambahan. AWS DMS mendukung Oracle ASM tanpa hak istimewa akun tambahan.	Lihat Hak istimewa akun tambahan yang diperlukan saat menggunakan Binary Reader dengan Oracle ASM .

Untuk langkah alur kerja ini	Jika Anda menggunakan CDC LogMiner, lakukan ini	Jika Anda melakukan CDC menggunakan Binary Reader, lakukan ini
Jika Anda belum melakukannya, konfigurasi tugas yang akan digunakan LogMiner atau Binary Reader untuk CDC.	Lihat Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC .	Lihat Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC .
Konfigurasi Oracle Standby sebagai sumber untuk CDC.	AWS DMS tidak mendukung Oracle Standby sebagai sumber.	Lihat Menggunakan Oracle Standby yang dikelola sendiri sebagai sumber dengan Binary Reader untuk CDC pada AWS DMS .

Gunakan langkah-langkah alur kerja berikut untuk mengonfigurasi instance database sumber Oracle yang AWS dikelola.

Untuk langkah alur kerja ini	Jika Anda menggunakan CDC LogMiner, lakukan ini	Jika Anda melakukan CDC menggunakan Binary Reader, lakukan ini
Berikan hak istimewa akun Oracle.	Untuk informasi selengkapnya, lihat Hak istimewa akun pengguna diperlukan pada sumber Oracle AWS yang dikelola untuk AWS DMS .	Untuk informasi selengkapnya, lihat Hak istimewa akun pengguna diperlukan pada sumber Oracle AWS yang dikelola untuk AWS DMS .
Siapkan basis data sumber untuk replikasi menggunakan CDC.	Untuk informasi selengkapnya, lihat Mengkonfigurasi sumber AWS Oracle yang dikelola untuk AWS DMS .	Untuk informasi selengkapnya, lihat Mengkonfigurasi sumber AWS Oracle yang dikelola untuk AWS DMS .
Berikan hak istimewa pengguna Oracle tambahan yang diperlukan untuk CDC.	Tidak perlu hak istimewa akun tambahan.	Untuk informasi selengkapnya, lihat Mengkonfigurasi tugas CDC untuk menggunakan

Untuk langkah alur kerja ini	Jika Anda menggunakan CDC LogMiner, lakukan ini	Jika Anda melakukan CDC menggunakan Binary Reader, lakukan ini
		Binary Reader dengan RDS untuk sumber Oracle AWS DMS.
Jika Anda belum melakukannya, konfigurasi tugas yang akan digunakan LogMiner atau Binary Reader untuk CDC.	Untuk informasi selengkapnya, lihat Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC.	Untuk informasi selengkapnya, lihat Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC.
Konfigurasi Oracle Standby sebagai sumber untuk CDC.	AWS DMS tidak mendukung Oracle Standby sebagai sumber.	Untuk informasi selengkapnya, lihat Menggunakan Amazon RDS Oracle Standby (replika baca) sebagai sumber dengan Binary Reader untuk CDC di AWS DMS.

Bekerja dengan database Oracle yang dikelola sendiri sebagai sumber AWS DMS

Suatu basis data yang dikelola sendiri adalah basis data yang Anda konfigurasi dan kontrol, baik instans basis data lokal atau basis data pada Amazon EC2. Berikut ini, Anda dapat mengetahui tentang hak istimewa dan konfigurasi yang Anda butuhkan saat menggunakan database Oracle yang dikelola sendiri. AWS DMS

Hak istimewa akun pengguna diperlukan pada sumber Oracle yang dikelola sendiri untuk AWS DMS

Untuk menggunakan database Oracle sebagai sumber AWS DMS, berikan hak istimewa berikut kepada pengguna Oracle yang ditentukan dalam pengaturan koneksi endpoint Oracle.

Note

Ketika memberikan hak istimewa, gunakan nama objek yang sebenarnya, bukan sinonim dari setiap objek. Misalnya, gunakan `V_$OBJECT` termasuk garis bawahnya, bukan `V$OBJECT` tanpa garis bawah.

```
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 Metode enkripsi
  yang didukung untuk menggunakan Oracle sebagai sumber 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.
```

Berikan hak istimewa tambahan berikut untuk setiap tabel yang direplikasi ketika Anda menggunakan daftar tabel tertentu.

```
GRANT SELECT on any-replicated-table to db_user;
```

Berikan hak istimewa tambahan berikut untuk memvalidasi kolom LOB dengan fitur validasi.

```
GRANT EXECUTE ON SYS.DBMS_CCRYPTO TO db_user;
```

Berikan hak istimewa tambahan berikut jika Anda menggunakan pembaca biner alih-alih. LogMiner

```
GRANT SELECT ON SYS.DBA_DIRECTORIES TO db_user;
```

Berikan hak istimewa tambahan berikut ini untuk menunjukkan tampilan.

```
GRANT SELECT on ALL_VIEWS to dms_user;
```

Untuk menunjukkan tampilan, Anda juga harus menambahkan atribut sambungan tambahan `exposeViews=true` ke titik akhir sumber Anda.

Berikan hak istimewa berikut tambahan saat menggunakan replikasi tanpa server.

```
GRANT SELECT on dba_segments to db_user;
```

Untuk informasi tentang replikasi tanpa server, lihat. [Bekerja dengan Tanpa AWS DMS Server](#)

Berikan hak istimewa tambahan berikut saat menggunakan penilaian premi khusus 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;  
GRANT SELECT ON DBA_USERS to dms_user;  
GRANT SELECT ON DBA_DIRECTORIES to dms_user;
```

Untuk informasi tentang penilaian premi khusus Oracle, lihat. [Penilaian Oracle](#)

Prasyarat untuk menangani transaksi terbuka untuk Oracle Standby

Saat menggunakan AWS DMS versi 3.4.6 dan yang lebih tinggi, lakukan langkah-langkah berikut untuk menangani transaksi terbuka untuk Oracle Standby.

1. Buat link database bernama, `AWSDMS_DBLINK` pada database utama. `DMS_USER` akan menggunakan link database untuk terhubung ke database utama. Perhatikan bahwa link database dijalankan dari instance standby untuk query transaksi terbuka yang berjalan pada database utama. Lihat contoh berikut ini.

```
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. Verifikasi koneksi ke link database menggunakan `DMS_USER` dibuat, seperti yang ditunjukkan pada contoh berikut.

```
select 1 from dual@AWSDMS_DBLINK
```

Mempersiapkan database sumber yang dikelola sendiri Oracle untuk CDC menggunakan AWS DMS

Siapkan basis data Oracle yang dikelola sendiri sebagai sumber untuk menjalankan tugas CDC dengan melakukan hal berikut:

- [Memverifikasi yang AWS DMS mendukung versi basis data sumber.](#)
- [Memastikan bahwa mode ARCHLOG aktif.](#)
- [Mengatur supplemental logging.](#)

Memverifikasi yang AWS DMS mendukung versi basis data sumber

Menjalankan kueri seperti berikut untuk melakukan verifikasi bahwa versi yang digunakan saat ini dari basis data sumber Oracle didukung oleh AWS DMS.

```
SELECT name, value, description FROM v$parameter WHERE name = 'compatible';
```

Di sini, `name`, `value`, dan `description` adalah kolom di suatu tempat di basis data yang sedang dijalankan kueri berdasarkan nilai `name`. Jika kueri ini berjalan tanpa kesalahan, AWS DMS mendukung versi database saat ini dan Anda dapat melanjutkan migrasi. Jika kueri menimbulkan kesalahan, AWS DMS tidak mendukung versi database saat ini. Untuk melanjutkan migrasi, pertama-tama konversikan database Oracle ke versi yang didukung oleh AWS DMS.

Memastikan bahwa mode ARCHIVELOG aktif

Anda dapat menjalankan Oracle dalam dua mode yang berbeda: mode ARCHIVELOG dan mode NOARCHIVELOG. Untuk menjalankan tugas CDC, jalankan basis data di mode ARCHIVELOG. Untuk mengetahui apakah database dalam ARCHIVELOG mode, jalankan query berikut.

```
SQL> SELECT log_mode FROM v$database;
```

Jika NOARCHIVELOG mode dikembalikan, atur database ARCHIVELOG sesuai instruksi Oracle.

Mengatur supplemental logging

Untuk menangkap perubahan yang sedang berlangsung, Anda AWS DMS mengharuskan Anda mengaktifkan logging tambahan minimal pada database sumber Oracle Anda. Selain itu, Anda perlu mengaktifkan supplemental logging pada setiap tabel yang direplikasi pada basis data.

Secara default, AWS DMS menambahkan logging PRIMARY KEY tambahan pada semua tabel yang direplikasi. AWS DMS Untuk memungkinkan menambahkan logging PRIMARY KEY tambahan, berikan hak istimewa berikut untuk setiap tabel yang direplikasi.

```
ALTER on any-replicated-table;
```

Anda dapat menonaktifkan logging PRIMARY KEY tambahan default yang ditambahkan dengan AWS DMS menggunakan atribut koneksi tambahan. `addSupplementalLogging` Untuk informasi selengkapnya, lihat [Pengaturan titik akhir saat menggunakan Oracle sebagai sumber AWS DMS](#).

Pastikan untuk mengaktifkan logging tambahan jika tugas replikasi Anda memperbarui tabel menggunakan WHERE klausa yang tidak mereferensikan kolom kunci utama.

Untuk mengatur supplemental logging secara manual

1. Menjalankan kueri berikut untuk melakukan verifikasi jika supplemental logging sudah diaktifkan untuk basis data.

```
SELECT supplemental_log_data_min FROM v$database;
```

Jika hasil yang dikembalikan adalah YES atau IMPLICIT, supplemental logging diaktifkan untuk basis data.

Jika tidak, aktifkan supplemental logging untuk basis data dengan menjalankan perintah berikut.

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

2. Pastikan bahwa supplemental logging yang diperlukan ditambahkan untuk setiap tabel yang direplikasi.

Pertimbangkan hal berikut:

- Jika supplemental logging ALL COLUMNS ditambahkan ke tabel, Anda tidak perlu menambah logging lagi.
- Jika kunci primer ada, tambahkan supplemental logging untuk kunci primer tersebut. Anda dapat melakukannya baik dengan menggunakan format untuk menambahkan supplemental logging pada kunci primer itu sendiri, atau dengan menambahkan supplemental logging pada kolom kunci primer pada basis data.

```
ALTER TABLE TableName ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;  
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
```

- Jika tidak ada kunci primer dan tabel memiliki indeks unik tunggal, tambahkan semua kolom indeks unik pada supplemental log.

```
ALTER TABLE TableName ADD SUPPLEMENTAL LOG GROUP LogGroupName  
(UniqueIndexColumn1 [, UniqueIndexColumn2] ...) ALWAYS;
```

Menggunakan SUPPLEMENTAL LOG DATA (UNIQUE INDEX) COLUMNS tidak menambahkan kolom indeks unik ke log.

- Jika tidak ada kunci primer dan tabel memiliki beberapa indeks unik, AWS DMS pilih indeks unik pertama dalam daftar naik yang diurutkan menurut abjad. Anda perlu menambahkan supplemental logging pada kolom indeks yang dipilih seperti pada item sebelumnya.

Menggunakan SUPPLEMENTAL LOG DATA (UNIQUE INDEX) COLUMNS tidak menambahkan kolom indeks unik pada log.

- Jika tidak ada kunci primer ada dan tidak ada indeks unik, tambahkan supplemental logging pada semua kolom.

```
ALTER TABLE TableName ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

Dalam beberapa kasus, kunci primer tabel atau indeks unik target berbeda dari kunci primer tabel atau indeks unik sumber. Dalam kasus tersebut, tambahkan supplemental logging secara manual pada kolom tabel sumber yang membentuk kunci primer tabel atau indeks unik target.

Dan jika Anda mengubah kunci primer tabel target, tambahkan supplemental logging pada kolom indeks unik target, bukan kolom kunci primer atau indeks unik sumber.

Jika filter atau transformasi didefinisikan untuk tabel, Anda mungkin perlu mengaktifkan logging tambahan.

Pertimbangkan hal berikut:

- Jika supplemental logging ALL COLUMNS ditambahkan pada tabel, Anda tidak perlu menambahkan logging lagi.
- Jika tabel memiliki indeks unik atau kunci primer, tambahkan tambahan supplemental logging pada setiap kolom yang terlibat dalam filter atau transformasi. Namun, lakukan hal tersebut hanya jika kolom-kolom itu berbeda dari kolom kunci primer atau indeks unik.
- Jika transformasi hanya mencakup satu kolom, jangan tambahkan kolom ini ke grup supplemental logging. Misalnya, untuk transformasi `A+B`, tambahkan supplemental logging pada kolom A dan kolom B. Namun, untuk transformasi `substring(A,10)` jangan tambahkan supplemental logging di kolom A.
- Untuk mengatur supplemental logging pada kolom kunci primer atau indeks unik kolom dan kolom lain yang disaring atau diubah, Anda dapat mengatur supplemental logging `USER_LOG_GROUP`.

Tambahkan logging ini pada kolom kunci primer atau indeks unik dan kolom tertentu lainnya yang disaring atau diubah.

Misalnya, untuk mereplikasi tabel bernama TEST.LOGGING dengan kunci primer ID dan filter oleh kolom NAME, Anda dapat menjalankan perintah yang mirip dengan berikut ini untuk membuat grup log supplemental logging.

```
ALTER TABLE TEST.LOGGING ADD SUPPLEMENTAL LOG GROUP TEST_LOG_GROUP (ID, NAME) ALWAYS;
```

Hak istimewa akun diperlukan saat menggunakan Oracle LogMiner untuk mengakses log redo

Untuk mengakses log redo menggunakan Oracle LogMiner, berikan hak istimewa berikut kepada pengguna Oracle yang ditentukan dalam pengaturan koneksi endpoint 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.
```

Hak istimewa akun diperlukan saat menggunakan AWS DMS Binary Reader untuk mengakses redo log

Untuk mengakses log redo menggunakan AWS DMS Binary Reader, berikan hak istimewa berikut kepada pengguna Oracle yang ditentukan dalam pengaturan koneksi endpoint 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 bekerja dengan fitur file Oracle yang mencakup direktori Oracle. Setiap objek direktori Oracle termasuk nama folder yang berisi file log redo untuk memproses. Direktori Oracle ini tidak diwakili pada tingkat sistem file. Sebaliknya, mereka adalah direktori logis yang dibuat pada tingkat basis data Oracle. Anda dapat melihatnya di gambaran ALL_DIRECTORIES Oracle.

Jika Anda AWS DMS ingin membuat direktori Oracle ini, berikan `CREATE ANY DIRECTORY` hak istimewa yang ditentukan sebelumnya. AWS DMS membuat nama direktori dengan `DMS_` awalan. Jika Anda tidak memberikan `CREATE ANY DIRECTORY` hak istimewa, buatlah direktori yang sesuai secara manual. Dalam beberapa kasus ketika Anda membuat direktori Oracle secara manual, pengguna Oracle yang ditentukan dalam titik akhir sumber Oracle bukanlah pengguna yang membuat direktori ini. Dalam kasus ini, berikan juga hak istimewa kepada `READ on DIRECTORY`.

Jika titik akhir sumber Oracle ada di Active Dataguard Standby (ADG), lihat [Cara menggunakan Binary Reader dengan posting ADG](#) di Blog Database. AWS

Note

AWS DMS CDC tidak mendukung Active Dataguard Standby yang tidak dikonfigurasi untuk menggunakan layanan transportasi redo otomatis.

Dalam beberapa kasus, Anda mungkin menggunakan Oracle Managed Files (OMF) untuk menyimpan log. Atau titik akhir sumber Anda ada di ADG dan dengan demikian Anda tidak dapat memberikan hak istimewa `BUAT DIREKTORI APAPUN`. Dalam kasus ini, buat direktori secara manual dengan semua lokasi log yang mungkin sebelum memulai tugas AWS DMS replikasi. Jika AWS DMS tidak menemukan direktori yang telah dibuat sebelumnya yang diharapkan, tugas berhenti. Dan, AWS DMS tidak menghapus entri yang telah dibuat di gambaran `ALL_DIRECTORIES`, jadi secara manual hapus semuanya.

Hak istimewa akun tambahan yang diperlukan saat menggunakan Binary Reader dengan Oracle ASM

Untuk mengakses log redo pada Automatic Storage Management (ASM) menggunakan Binary Reader, berikan hak istimewa berikut untuk pengguna Oracle yang ditentukan pada pengaturan sambungan titik akhir 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.
```

Anda dapat melakukan validasi akses akun ASM dengan membuka prompt perintah dan meminta salah satu pernyataan berikut, tergantung pada versi Oracle Anda sebagaimana ditentukan sebelumnya.

Jika Anda membutuhkan hak istimewa SYSDBA, gunakan berikut ini.

```
sqlplus asmuser/asmpassword@+asmserver as sysdba
```

Jika Anda membutuhkan hak istimewa SYSASM, gunakan berikut ini.

```
sqlplus asmuser/asmpassword@+asmserver as sysasm
```

Menggunakan Oracle Standby yang dikelola sendiri sebagai sumber dengan Binary Reader untuk CDC pada AWS DMS

Untuk mengonfigurasi instans Oracle Standby sebagai sumber ketika menggunakan Binary Reader untuk CDC, mulai dengan prasyarat berikut:

- AWS DMS saat ini hanya mendukung Oracle Active Data Guard Standby.
- Pastikan bahwa konfigurasi Oracle Data Guard menggunakan:
 - Pengulangan layanan transportasi untuk transfer otomatis data pengulangan.
 - Terapkan layanan untuk secara otomatis menerapkan pengulangan ke basis data standby.

Untuk mengonfirmasi persyaratan tersebut terpenuhi, jalankan kueri berikut.

```
SQL> select open_mode, database_role from v$database;
```

Dari output kueri itu, konfirmasi bahwa database siaga dibuka dalam mode READ ONLY dan redo diterapkan secara otomatis. Misalnya:

```
OPEN_MODE          DATABASE_ROLE
-----          -
READ ONLY WITH APPLY  PHYSICAL STANDBY
```

Untuk mengonfigurasi instans Oracle Standby sebagai sumber ketika menggunakan Binary Reader untuk CDC

1. Berikan hak istimewa tambahan yang diperlukan untuk mengakses berkas log standby.

```
GRANT SELECT ON v_$standby_log TO db_user;
```

2. Buat titik akhir sumber untuk Oracle Standby dengan menggunakan AWS Management Console atau AWS CLI. Saat membuat titik akhir, tentukan atribut sambungan tambahan berikut.

```
useLogminerReader=N;useBfile=Y;
```

Note

Di AWS DMS, Anda dapat menggunakan atribut koneksi tambahan untuk menentukan apakah Anda ingin bermigrasi dari log arsip alih-alih log pengulangan. Untuk informasi selengkapnya, lihat [Pengaturan titik akhir saat menggunakan Oracle sebagai sumber AWS DMS](#).

3. Konfigurasi tujuan log yang diarsipkan.

Pembaca biner DMS untuk sumber Oracle tanpa ASM menggunakan Direktori Oracle untuk mengakses log redo yang diarsipkan. Jika database Anda dikonfigurasi untuk menggunakan Fast Recovery Area (FRA) sebagai tujuan log arsip, lokasi file redo arsip tidak konstan. Setiap hari redo log yang diarsipkan dihasilkan menghasilkan direktori baru yang dibuat di FRA, menggunakan format nama direktori YYYY_MM_DD. Misalnya:

```
DB_RECOVERY_FILE_DEST/SID/archivelog/YYYY_MM_DD
```

Ketika DMS membutuhkan akses ke file redo yang diarsipkan di direktori FRA yang baru dibuat dan basis data baca-tulis utama digunakan sebagai sumber, DMS membuat direktori Oracle baru atau menggantikan direktori Oracle yang ada, sebagai berikut.

```
CREATE OR REPLACE DIRECTORY dmsrep_taskid AS 'DB_RECOVERY_FILE_DEST/SID/archivelog/  
YYYY_MM_DD';
```

Ketika database siaga digunakan sebagai sumber, DMS tidak dapat membuat atau mengganti direktori Oracle karena database dalam mode read-only. Namun, Anda dapat memilih untuk melakukan salah satu langkah tambahan ini:

- a. Ubah `log_archive_dest_id_1` untuk menggunakan jalur aktual alih-alih FRA dalam konfigurasi sedemikian rupa sehingga Oracle tidak akan membuat subdirektori harian:

```
ALTER SYSTEM SET log_archive_dest_1='LOCATION=full directory path'
```

Kemudian, buat objek direktori Oracle yang akan digunakan oleh DMS:

```
CREATE OR REPLACE DIRECTORY dms_archived_logs AS 'full directory path';
```

- b. Buat tujuan log arsip tambahan dan objek direktori Oracle yang menunjuk ke tujuan itu. Misalnya:

```
ALTER SYSTEM SET log_archive_dest_3='LOCATION=full directory path';  
CREATE DIRECTORY dms_archived_log AS 'full directory path';
```

Kemudian tambahkan atribut koneksi tambahan ke titik akhir sumber tugas:

```
archivedLogDestId=3
```

- c. Secara manual membuat objek direktori Oracle untuk digunakan oleh 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. Buat pekerjaan penjadwal Oracle yang berjalan setiap hari dan membuat direktori yang diperlukan.

Menggunakan database yang dikelola pengguna di Oracle Cloud Infrastructure (OCI) sebagai sumber untuk CDC di AWS DMS

Database yang dikelola pengguna adalah database yang Anda konfigurasi dan kontrol, seperti database Oracle yang dibuat pada mesin virtual (VM), bare metal, atau server Exadata. Atau, database yang Anda konfigurasi dan kontrol yang berjalan pada infrastruktur khusus, seperti Oracle Cloud Infrastructure (OCI). Informasi berikut menjelaskan hak istimewa dan konfigurasi yang Anda butuhkan saat menggunakan database yang dikelola pengguna Oracle di OCI sebagai sumber untuk pengambilan data perubahan (CDC) di AWS DMS

Untuk mengonfigurasi database Oracle yang dikelola pengguna yang dihosting OCI sebagai sumber untuk pengambilan data perubahan

1. Berikan hak istimewa akun pengguna yang diperlukan untuk database sumber Oracle yang dikelola pengguna di OCI. Untuk informasi selengkapnya, lihat [Hak istimewa akun untuk titik akhir sumber Oracle yang dikelola sendiri](#).
2. Berikan hak istimewa akun yang diperlukan saat menggunakan Binary Reader untuk mengakses log redo. Untuk informasi selengkapnya, lihat [Hak istimewa akun yang diperlukan saat menggunakan Pembaca Biner](#).
3. Tambahkan hak istimewa akun yang diperlukan saat menggunakan Binary Reader dengan Oracle Automatic Storage Management (ASM). Untuk informasi selengkapnya, lihat [Hak istimewa akun tambahan yang diperlukan saat menggunakan Binary Reader dengan Oracle ASM](#).
4. Siapkan penebangan tambahan. Untuk informasi selengkapnya, lihat [Menyiapkan pencatatan tambahan](#).
5. Set-up enkripsi TDE. Untuk informasi selengkapnya, lihat [Metode enkripsi saat menggunakan database Oracle sebagai titik akhir sumber](#).

Keterbatasan berikut berlaku saat mereplikasi data dari database sumber Oracle di Oracle Cloud Infrastructure (OCI).

Batasan

- DMS tidak mendukung penggunaan Oracle LogMiner untuk mengakses log redo.
- DMS tidak mendukung Autonomous DB.

Bekerja dengan database Oracle yang AWS dikelola sebagai sumber AWS DMS

Database yang AWS dikelola adalah database yang ada di layanan Amazon seperti Amazon RDS, Amazon Aurora, atau Amazon S3. Berikut ini, Anda dapat menemukan hak istimewa dan konfigurasi yang perlu Anda atur saat menggunakan database Oracle AWS-managed. AWS DMS

Hak istimewa akun pengguna diperlukan pada sumber Oracle AWS yang dikelola untuk AWS DMS

Berikan hak istimewa berikut kepada akun pengguna Oracle yang ditentukan dalam definisi titik akhir sumber Oracle.

⚠ Important

Untuk semua nilai parameter seperti *db_user* dan *any-replicated-table*, Oracle mengasumsikan nilai adalah semua huruf besar kecuali jika Anda menentukan nilai dengan pengidentifikasi kepekaan kapital. Misalnya, anggaplah Anda membuat nilai *db_user* tanpa menggunakan tanda petik, seperti dalam `CREATE USER myuser` atau `CREATE USER MYUSER`. Dalam hal ini, Oracle mengidentifikasi dan menyimpan nilai dalam huruf besar semua (MYUSER). Jika Anda menggunakan tanda petik, seperti dalam `CREATE USER "MyUser"` atau `CREATE USER 'MyUser'`, Oracle mengidentifikasi dan menyimpan nilai dengan kepekaan kapital yang Anda tentukan (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.
```

Selain itu, hiberikan izin SELECT dan EXECUTE pada objek SYS menggunakan prosedur Amazon RDS `rdsadmin.rdsadmin_util.grant_sys_object` seperti yang telah ditunjukkan. Untuk informasi selengkapnya, lihat [Memberikan hak istimewa PILIH atau JALANKAN pada objek 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_CCRYPTO', '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');
```

Untuk informasi selengkapnya tentang penggunaan Amazon RDS Active Dataguard Standby (ADG) dengan AWS DMS, lihat [Menggunakan Amazon RDS Oracle Standby \(replika baca\) sebagai sumber dengan Binary Reader untuk CDC di AWS DMS](#).

Untuk informasi lebih lanjut tentang menggunakan Oracle TDE dengan AWS DMS, lihat [Metode enkripsi yang didukung untuk menggunakan Oracle sebagai sumber AWS DMS](#)

Prasyarat untuk menangani transaksi terbuka untuk Oracle Standby

Saat menggunakan AWS DMS versi 3.4.6 dan yang lebih tinggi, lakukan langkah-langkah berikut untuk menangani transaksi terbuka untuk Oracle Standby.

1. Buat link database bernama, `AWSDMS_DBLINK` pada database utama. `DMS_USER` akan menggunakan link database untuk terhubung ke database utama. Perhatikan bahwa link database dijalankan dari instance standby untuk query transaksi terbuka yang berjalan pada database utama. Lihat contoh berikut ini.

```
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. Verifikasi koneksi ke link database menggunakan `DMS_USER` dibuat, seperti yang ditunjukkan pada contoh berikut.

```
select 1 from dual@AWSDMS_DBLINK
```

Mengkonfigurasi sumber AWS Oracle yang dikelola untuk AWS DMS

Sebelum menggunakan database Oracle yang AWS dikelola sebagai sumber AWS DMS, lakukan tugas-tugas berikut untuk database Oracle:

- Aktifkan backup otomatis Untuk informasi selengkapnya tentang mengaktifkan pencadangan otomatis, lihat [Mengaktifkan pencadangan otomatis](#) di Panduan Pengguna Amazon RDS.
- Siapkan supplemental logging
- Siapkan pengarsipan. Mengarsipkan log redo untuk instans Amazon RDS for Oracle DB AWS DMS memungkinkan untuk mengambil informasi log menggunakan Oracle atau Binary Reader. LogMiner

Mengatur pengarsipan

1. Jalankan perintah `rdsadmin.rdsadmin_util.set_configuration` untuk mengatur pengarsipan.

Misalnya, untuk mempertahankan log redo diarsipkan selama 24 jam, jalankan perintah berikut.

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24);
commit;
```

Note

Fungsi `commit` diperlukan agar perubahan dapat berpengaruh.

2. Pastikan penyimpanan Anda memiliki cukup ruang untuk log redo yang diarsipkan selama periode penyimpanan yang ditentukan. Misalnya, jika periode penyimpanan Anda adalah 24 jam, hitung ukuran total log redo yang diarsipkan yang terakumulasi selama durasi jam umumnya untuk pemrosesan transaksi dan kalikan totalnya dengan 24. Bandingkan total 24 jam yang dihitung ini dengan ruang penyimpanan yang tersedia dan tentukan apakah Anda memiliki cukup ruang penyimpanan untuk menangani pemrosesan transaksi 24 jam penuh.

Mengatur supplemental logging

1. Jalankan perintah berikut untuk mengaktifkan supplemental logging pada tingkat basis data.

```
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD');
```

2. Jalankan perintah berikut untuk mengaktifkan supplemental logging kunci primer.

```
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD','PRIMARY KEY');
```

3. (Opsional) Aktifkan supplemental logging tingkat kunci pada tingkat tabel.

Basis data sumber Anda menimbulkan sedikit overhead ketika supplemental logging tingkat kunci diaktifkan. Oleh karena itu, jika Anda memigrasikan hanya subset dari tabel Anda, Anda mungkin ingin mengaktifkan tingkat kunci supplemental logging pada tingkat tabel. Untuk mengaktifkan supplemental logging tingkat kunci pada tingkat tabel, jalankan perintah berikut.

```
alter table table_name add supplemental log data (PRIMARY KEY) columns;
```

Mengonfigurasi tugas CDC untuk menggunakan Binary Reader dengan RDS untuk sumber Oracle AWS DMS

Anda dapat mengonfigurasi AWS DMS untuk mengakses sumber Amazon RDS for Oracle instance redo log menggunakan Binary Reader untuk CDC.

Note

Untuk menggunakan Oracle LogMiner, hak istimewa akun pengguna minimum yang diperlukan sudah cukup. Untuk informasi selengkapnya, lihat [Hak istimewa akun pengguna diperlukan pada sumber Oracle AWS yang dikelola untuk AWS DMS](#).

Untuk menggunakan AWS DMS Binary Reader, tentukan pengaturan tambahan dan atribut koneksi tambahan untuk titik akhir sumber Oracle, tergantung pada versi Anda AWS DMS .

Dukungan Binary Reader tersedia pada versi Amazon RDS for Oracle berikut ini:

- Oracle 11.2 - Versi 11.2.0.4V11 dan lebih tinggi
- Oracle 12.1 - Versi 12.1.0.2.V7 dan lebih tinggi
- Oracle 12.2 - Semua versi
- Oracle 18.0 - Semua versi
- Oracle 19.0 - Semua versi

Mengonfigurasi CDC menggunakan Binary Reader

1. Masuk ke basis data sumber Amazon RDS for Oracle Anda sebagai pengguna utama dan jalankan prosedur yang tersimpan berikut untuk membuat direktori tingkat server.

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;  
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

2. Berikan hak istimewa berikut kepada akun pengguna Oracle yang digunakan untuk mengakses titik akhir sumber Oracle.

```
GRANT READ ON DIRECTORY ONLINELOG_DIR TO db_user;  
GRANT READ ON DIRECTORY ARCHIVELOG_DIR TO db_user;
```

3. Atur atribut sambungan tambahan berikut pada titik akhir sumber Oracle Amazon RDS:

- Untuk RDS Oracle versi 11.2 dan 12.1, atur berikut ini.

```
useLogminerReader=N;useBfile=Y;accessAlternateDirectly=false;useAlternateFolderForOnline=  
oraclePathPrefix=/rdsdbdata/db/{$DATABASE_NAME}_A/;usePathPrefix=/rdsdbdata/  
log/;replacePathPrefix=true;
```

- Untuk RDS Oracle versi 12.2, 18.0, dan 19.0, atur berikut ini.

```
useLogminerReader=N;useBfile=Y;
```

Note

Pastikan tidak ada spasi putih yang mengikuti pemisah titik koma (;) untuk beberapa pengaturan atribut, misalnya `oneSetting;thenAnother`.

Untuk informasi lebih lanjut tentang mengonfigurasi tugas CDC, lihat [Konfigurasi untuk CDC pada sumber basis data Oracle](#).

Menggunakan Amazon RDS Oracle Standby (replika baca) sebagai sumber dengan Binary Reader untuk CDC di AWS DMS

Lakukan verifikasi prasyarat berikut untuk menggunakan Amazon RDS for Oracle Standby sebagai sumber ketika menggunakan Binary Reader untuk CDC di AWS DMS:

- Gunakan pengguna utama Oracle untuk mengatur Binary Reader.
- Pastikan bahwa AWS DMS saat ini mendukung hanya menggunakan Oracle Active Data Guard Standby.

Setelah Anda melakukannya, gunakan prosedur berikut untuk menggunakan RDS for Oracle Standby sebagai sumber ketika menggunakan Binary Reader untuk CDC.

Mengonfigurasi RDS for Oracle Standby sebagai sumber saat menggunakan Binary Reader untuk CDC

1. Masuk ke RDS untuk instance utama Oracle sebagai pengguna utama.
2. Jalankan prosedur yang tersimpan berikut seperti yang didokumentasikan di panduan pengguna Amazon RDS untuk membuat direktori tingkat server.

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

3. Identifikasi direktori yang dibuat pada langkah 2.

```
SELECT directory_name, directory_path FROM all_directories
WHERE directory_name LIKE ( 'ARCHIVELOG_DIR_%' )
      OR directory_name LIKE ( 'ONLINELOG_DIR_%' )
```

Misalnya, kode sebelumnya menampilkan daftar direktori seperti berikut ini.

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. Berikan hak istimewa Read mengenai direktori sebelumnya kepada akun pengguna Oracle yang digunakan untuk mengakses 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. Lakukan penukaran log arsip pada instans utama. Dengan melakukannya, perubahan pada ALL_DIRECTORIES dapat dipastikan juga dipindahkan ke Oracle Standby.
6. Jalankan kueri ALL_DIRECTORIES pada Oracle Standby untuk mengonfirmasi bahwa perubahan telah diterapkan.

7. Buat titik akhir sumber untuk Oracle Standby dengan menggunakan AWS DMS Management Console atau AWS Command Line Interface ().AWS CLI Sembari membuat titik akhir, tentukan atribut sambungan tambahan berikut.

```
useLogminerReader=N;useBfile=Y;archivedLogDestId=1;additionalArchivedLogDestId=2
```

8. Setelah membuat titik akhir, gunakan koneksi titik akhir Uji pada halaman Buat titik akhir konsol atau AWS CLI `test-connection` perintah untuk memverifikasi bahwa konektivitas dibuat.

Keterbatasan dalam menggunakan Oracle sebagai sumber AWS DMS

Keterbatasan berikut berlaku saat menggunakan basis data Oracle sebagai sumber untuk AWS DMS:

- AWS DMS mendukung tipe data Oracle Extended dalam AWS DMS versi 3.5.0 dan yang lebih tinggi.
- AWS DMS tidak mendukung nama objek yang panjang (lebih dari 30 byte).
- AWS DMS tidak mendukung indeks berbasis fungsi.
- Jika Anda mengelola supplemental logging dan melakukan transformasi pada salah satu kolom, pastikan bahwa supplemental logging diaktifkan untuk semua bidang dan kolom. Untuk informasi selengkapnya tentang pengaturan supplemental logging, lihat topik berikut:
 - Untuk basis data sumber Oracle yang dikelola sendiri, lihat [Mengatur supplemental logging](#).
 - Untuk database sumber Oracle yang AWS dikelola, lihat. [Mengkonfigurasi sumber AWS Oracle yang dikelola untuk AWS DMS](#)
- AWS DMS tidak mendukung database root kontainer multi-penyewa (CDB\$ROOT). DMS mendukung PDB menggunakan Binary Reader.
- AWS DMS tidak mendukung kendala yang ditangguhkan.
- LOB aman didukung menggunakan mode LOB Penuh hanya dengan melakukan pencarian LOB.
- AWS DMS mendukung `rename table table-name to new-table-name` sintaks untuk semua versi Oracle yang didukung 11 dan lebih tinggi. Sintaks ini tidak didukung untuk sumber basis data Oracle versi 10.
- AWS DMS tidak mereplikasi hasil pernyataan DDL. `ALTER TABLE ADD column data_type DEFAULT default_value` Alih-alih mereplikasi *default_value* ke target, DMS menetapkan kolom baru untuk NULL.

- Saat menggunakan AWS DMS versi 3.4.7 atau lebih tinggi, untuk mereplikasi perubahan yang dihasilkan dari operasi partisi atau subpartisi, lakukan hal berikut sebelum memulai tugas DMS.
 - Buat struktur tabel yang dipartisi (DDL) secara manual;
 - Pastikan DDL sama pada sumber Oracle dan target Oracle;
 - Mengatur atribut koneksi tambahan `enableHomogenousPartitionOps=true`.

Untuk informasi selengkapnya tentang `enableHomogenousPartitionOps`, lihat [Pengaturan titik akhir saat menggunakan Oracle sebagai sumber AWS DMS](#). Juga, perhatikan bahwa pada tugas FULL+CDC, DMS tidak mereplikasi perubahan data yang ditangkap sebagai bagian dari perubahan cache. Dalam kasus penggunaan itu, buat ulang struktur tabel pada target Oracle dan muat ulang tabel yang dimaksud.

Sebelum AWS DMS versi 3.4.7:

DMS tidak mereplikasi perubahan data yang dihasilkan dari operasi partisi atau subpartisi (ADD,, DROPEXCHANGE, dan). TRUNCATE Pembaruan tersebut dapat menyebabkan kesalahan berikut selama replikasi:

- Untuk operasi ADD, pembaruan dan penghapusan data tambahan mungkin memunculkan peringatan “0 baris terpengaruh”.
- Untuk operasi DROP dan TRUNCATE, sisipan baru mungkin memunculkan kesalahan “duplikat”.
- Operasi EXCHANGE mungkin memunculkan peringatan kesalahan “0 baris terpengaruh” dan “duplikat”.

Untuk mereplikasi perubahan yang dihasilkan dari operasi partisi atau subpartisi, load ulang tabel yang bersangkutan. Setelah menambahkan partisi kosong baru, operasi pada partisi yyang baru ditambahkan direplikasi ke target seperti biasa.

- AWS DMS versi sebelum 3.4 tidak mendukung perubahan data pada target yang dihasilkan dari menjalankan `CREATE TABLE AS` pernyataan pada sumber. Namun, tabel baru dibuat pada target.
- AWS DMS tidak menangkap perubahan yang dibuat oleh `DBMS_REDEFINITION` paket Oracle, misalnya metadata tabel dan bidang. `OBJECT_ID`
- AWS DMS memetakan kolom BLOB dan CLOB NULL kosong ke target.
- Saat menangkap perubahan dengan Oracle 11 LogMiner, pembaruan pada kolom CLOB dengan panjang string lebih besar dari 1982 hilang, dan target tidak diperbarui.
- Selama pengambilan data perubahan (CDC), AWS DMS tidak mendukung pembaruan batch ke kolom numerik yang didefinisikan sebagai kunci utama.

- AWS DMS tidak mendukung UPDATE perintah tertentu. Contoh berikut adalah perintah UPDATE yang tidak didukung.

```
UPDATE TEST_TABLE SET KEY=KEY+1;
```

Di sini, TEST_TABLE adalah nama tabel dan KEY adalah kolom numerik yang didefinisikan sebagai kunci primer.

- AWS DMS tidak mendukung mode LOB penuh untuk memuat kolom LONG dan LONG RAW. Sebagai gantinya, Anda dapat menggunakan mode LOB terbatas untuk memigrasikan tipe data ini ke target Oracle. Dalam mode LOB terbatas, AWS DMS potong data apa pun menjadi 64 KB yang Anda atur ke kolom LONG atau LONG RAW yang lebih panjang dari 64 KB.
- AWS DMS tidak mendukung mode LOB penuh untuk memuat kolom XMLTYPE. Sebagai gantinya, Anda dapat menggunakan mode LOB terbatas untuk memigrasikan kolom XMLTYPE ke target Oracle. Dalam mode LOB terbatas, DMS memotong data yang lebih besar dari variabel 'Ukuran LOB maksimum' yang ditentukan pengguna. Nilai maksimum yang disarankan untuk 'Ukuran LOB maksimum' adalah 100MB.
- AWS DMS tidak mereplikasi tabel yang namanya mengandung apostrof.
- AWS DMS mendukung CDC dari pandangan yang terwujud. Tetapi DMS tidak mendukung CDC dari pandangan lain.
- AWS DMS tidak mendukung CDC untuk tabel yang terorganisir indeks dengan segmen overflow.
- AWS DMS tidak mendukung Drop Partition operasi untuk tabel yang dipartisi dengan referensi dengan enableHomogenousPartitionOps set ke. true
- Bila Anda menggunakan Oracle LogMiner untuk mengakses redo log, AWS DMS memiliki batasan berikut:
 - Hanya untuk Oracle 12, AWS DMS tidak mereplikasi perubahan apa pun pada kolom LOB.
 - Untuk semua versi Oracle, AWS DMS tidak mereplikasi hasil UPDATE operasi pada XMLTYPE dan kolom LOB.
 - AWS DMS tidak mendukung transaksi XA dalam replikasi saat menggunakan Oracle. LogMiner
 - Oracle LogMiner tidak mendukung koneksi ke database pluggable (PDB). Untuk menyambung ke PDB, akses log redo menggunakan Binary Reader.
 - Operasi SHRINK SPACE tidak didukung.
- Bila Anda menggunakan Binary Reader, AWS DMS memiliki keterbatasan ini:
 - DMS tidak mendukung klaster tabel.

- DMS mendukung hanya operasi SHRINK SPACE tingkat tabel. Level ini mencakup tabel lengkap, partisi, dan sub-partisi.
- Level ini tidak mendukung perubahan ke tabel yang terorganisir secara indeks dengan kompresi kunci.
- Level ini tidak mendukung penerapan log redo online pada perangkat mentah.
- Binary Reader mendukung TDE hanya untuk database Oracle yang dikelola sendiri karena RDS untuk Oracle tidak mendukung pengambilan kata sandi dompet untuk kunci enkripsi TDE.
- AWS DMS tidak mendukung koneksi ke sumber Amazon RDS Oracle menggunakan proxy Oracle Automatic Storage Management (ASM).
- AWS DMS tidak mendukung kolom virtual.
- AWS DMS tidak mendukung tipe ROWID data atau tampilan terwujud berdasarkan kolom ROWID.

AWS DMS memiliki sebagian dukungan untuk Oracle Materialized Views. Untuk muatan penuh, DMS dapat melakukan salinan penuh Oracle Materialized View. DMS menyalin Tampilan Materialisasi sebagai tabel dasar ke sistem target dan mengabaikan kolom ROWID apa pun di Tampilan Materialisasi. Untuk replikasi berkelanjutan (CDC), DMS mencoba mereplikasi perubahan pada data Tampilan Materialisasi tetapi hasilnya mungkin tidak ideal. Secara khusus, jika Tampilan Materialisasi benar-benar disegarkan, DMS mereplikasi penghapusan individual untuk semua baris, diikuti oleh sisipan individual untuk semua baris. Itu adalah latihan yang sangat intensif sumber daya dan mungkin berkinerja buruk untuk tampilan terwujud dengan sejumlah besar baris. Untuk replikasi berkelanjutan di mana tampilan terwujud melakukan penyegaran cepat, DMS mencoba memproses dan mereplikasi perubahan data penyegaran cepat. Dalam kedua kasus, DMS melewatkan kolom ROWID apa pun dalam tampilan terwujud.

- AWS DMS tidak memuat atau menangkap tabel sementara global.
- Untuk target S3 yang menggunakan replikasi, aktifkan supplemental logging pada setiap kolom sehingga sumber pembaruan baris dapat menangkap setiap nilai kolom. Lihat contoh berikut:
`alter table yourtablename add supplemental log data (all) columns;`
- Pembaruan untuk baris dengan kunci unik komposit yang berisi tidak null dapat direplikasi pada target.
- AWS DMS tidak mendukung penggunaan beberapa kunci enkripsi Oracle TDE pada titik akhir sumber yang sama. Setiap titik akhir hanya dapat memiliki satu atribut untuk enkripsi TDE Nama Kunci "securityDbEncryptionName", dan satu kata sandi TDE untuk kunci ini.
- Saat mereplikasi dari Amazon RDS for Oracle, TDE hanya didukung dengan tablespace terenkripsi dan menggunakan Oracle. LogMiner

- AWS DMS tidak mendukung beberapa operasi penggantian nama tabel secara berurutan.
- Saat menggunakan Oracle 19.0 sebagai sumber, AWS DMS tidak mendukung fitur berikut:
 - Pengalihan DML-penjaga data
 - Tabel hibrida yang dipartisi
 - Akun Oracle khusus skema
- AWS DMS tidak mendukung migrasi tabel atau tampilan jenis BIN\$ atauDR\$.
- Dimulai dengan Oracle 18.x, AWS DMS tidak mendukung pengambilan data perubahan (CDC) dari Oracle Express Edition (Oracle Database XE).
- Saat memigrasikan data dari kolom CHAR, DMS memotong spasi tambahan apa pun.
- AWS DMS tidak mendukung replikasi dari wadah aplikasi.
- AWS DMS tidak mendukung kinerja Oracle Flashback Database dan restore point, karena operasi ini memengaruhi konsistensi file Oracle Redo Log.
- INSERTProsedur pemuatan langsung dengan opsi eksekusi paralel tidak didukung dalam kasus berikut:
 - Tabel tidak terkompresi dengan lebih dari 255 kolom
 - Ukuran baris melebihi 8K
 - Tabel Exadata HCC
 - Database berjalan di platform Big Endian
- Tabel sumber dengan kunci primer atau unik tidak memerlukan semua pencatatan tambahan COLUMN untuk diaktifkan. Ini menciptakan lebih banyak aktivitas redo log dan dapat meningkatkan latensi DMS CDC.
- AWS DMS tidak memigrasikan data dari kolom tak terlihat di database sumber Anda. Untuk menyertakan kolom ini dalam lingkup migrasi Anda, gunakan ALTER TABLE pernyataan untuk membuat kolom ini terlihat.

Dukungan SSL untuk titik akhir Oracle

AWS DMS Endpoint Oracle mendukung SSL V3 untuk mode dan SSL. none verify-ca Untuk menggunakan SSL dengan titik akhir Oracle, unggah Oracle wallet untuk titik akhir, bukannya file sertifikat .pem.

Topik

- [Menggunakan sertifikat yang ada untuk Oracle SSL](#)

- [Menggunakan sertifikat yang ditandatangani sendiri untuk Oracle SSL](#)

Menggunakan sertifikat yang ada untuk Oracle SSL

Untuk menggunakan instalasi klien Oracle yang ada untuk membuat file Oracle wallet dari file sertifikat CA, lakukan langkah-langkah berikut.

Untuk menggunakan instalasi klien oracle yang ada untuk Oracle SSL dengan AWS DMS

1. Mengatur variabel sistem ORACLE_HOME ke lokasi direktori dbhome_1 Anda dengan menjalankan perintah berikut.

```
prompt>export ORACLE_HOME=/home/user/app/user/product/12.1.0/dbhome_1
```

2. Tambahkan \$ORACLE_HOME/lib ke variabel sistem LD_LIBRARY_PATH.

```
prompt>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

3. Buat direktori untuk Oracle wallet di \$ORACLE_HOME/ssl_wallet.

```
prompt>mkdir $ORACLE_HOME/ssl_wallet
```

4. Tempatkan file .pem sertifikat CA dalam direktori ssl_wallet. Jika Anda menggunakan Amazon RDS, Anda dapat mengunduh file sertifikat CA root rds-ca-2015-root.pem yang di-host oleh Amazon RDS. Untuk informasi selengkapnya tentang pengunduhan file ini, lihat [Menggunakan SSL/TLS untuk mengenkripsi sambungan ke instans DB](#) dalam Panduan Pengguna Amazon RDS.
5. Jalankan perintah berikut untuk membuat Oracle wallet.

```
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
```

Ketika Anda telah menyelesaikan langkah-langkah sebelumnya, Anda dapat mengimpor file wallet dengan `ImportCertificate` panggilan API dengan menentukan parameter sertifikat-wallet. Anda kemudian dapat menggunakan sertifikat wallet yang diimpor ketika Anda memilih `verify-ca` sebagai mode SSL saat membuat atau memodifikasi titik akhir Oracle Anda.

Note

Dompot Oracle adalah file biner. AWS DMS menerima file-file ini apa adanya.

Menggunakan sertifikat yang ditandatangani sendiri untuk Oracle SSL

Untuk menggunakan sertifikat yang ditandatangani sendiri untuk Oracle SSL, lakukan langkah-langkah berikut, dengan asumsi kata sandi dompet Oracle. `oracle123`

Untuk menggunakan sertifikat yang ditandatangani sendiri untuk Oracle SSL dengan AWS DMS

1. Buat direktori yang akan Anda gunakan untuk bekerja dengan sertifikat yang ditandatangani sendiri.

```
mkdir -p /u01/app/oracle/self_signed_cert
```

2. Ubah ke direktori yang Anda buat pada langkah sebelumnya.

```
cd /u01/app/oracle/self_signed_cert
```

3. Buat kunci root.

```
openssl genrsa -out self-rootCA.key 2048
```

4. Tandatangi sendiri sertifikat root menggunakan kunci root yang Anda buat pada langkah sebelumnya.

```
openssl req -x509 -new -nodes -key self-rootCA.key  
-sha256 -days 3650 -out self-rootCA.pem
```

Gunakan parameter input seperti berikut ini.

- Country Name (2 letter code) [XX], misalnya: AU
- State or Province Name (full name) [], misalnya: NSW

- Locality Name (e.g., city) [Default City], misalnya: Sydney
- Organization Name (e.g., company) [Default Company Ltd], misalnya: AmazonWebService
- Organizational Unit Name (e.g., section) [], misalnya: DBeng
- Common Name (e.g., your name or your server's hostname) [], misalnya: aws
- Email Address [], misalnya: abcd.efgh@amazonwebservice.com

5. Buat direktori untuk Oracle wallet di basis data Oracle.

```
mkdir -p /u01/app/oracle/wallet
```

6. Membuat Oracle basis data baru.

```
orapki wallet create -wallet "/u01/app/oracle/wallet" -pwd oracle123 -  
auto_login_local
```

7. Tambahkan sertifikat root ke Oracle wallet.

```
orapki wallet add -wallet "/u01/app/oracle/wallet" -pwd oracle123 -trusted_cert  
-cert /u01/app/oracle/self_signed_cert/self-rootCA.pem
```

8. Daftar isi dari Oracle wallet. Daftar harus mencakup sertifikat root.

```
orapki wallet display -wallet /u01/app/oracle/wallet -pwd oracle123
```

Misalnya, ini mungkin ditampilkan mirip dengan yang berikut ini.

```
Requested Certificates:  
User Certificates:  
Trusted Certificates:  
Subject:          CN=aws,OU=DBeng,O= AmazonWebService,L=Sydney,ST=NSW,C=AU
```

9. Buat Certificate Signing Request (CSR) menggunakan utilitas ORAPKI.

```
orapki wallet add -wallet "/u01/app/oracle/wallet" -pwd oracle123  
-dn "CN=aws" -keysize 2048 -sign_alg sha256
```

10. Jalankan perintah berikut.

```
openssl pkcs12 -in /u01/app/oracle/wallet/ewallet.p12 -nodes -out /u01/app/oracle/wallet/nonoracle_wallet.pem
```

Ini memiliki output seperti berikut ini.

```
Enter Import Password:  
MAC verified OK  
Warning unsupported bag type: secretBag
```

11. Masukan 'dms' sebagai nama umum.

```
openssl req -new -key /u01/app/oracle/wallet/nonoracle_wallet.pem -out certdms.csr
```

Gunakan parameter input seperti berikut ini.

- Country Name (2 letter code) [XX], misalnya: AU
- State or Province Name (full name) [], misalnya: NSW
- Locality Name (e.g., city) [Default City], misalnya: Sydney
- Organization Name (e.g., company) [Default Company Ltd], misalnya: AmazonWebService
- Organizational Unit Name (e.g., section) [], misalnya: aws
- Common Name (e.g., your name or your server's hostname) [], misalnya: aws
- Email Address [], misalnya: abcd.efgh@amazonwebservice.com

Pastikan ini tidak sama dengan langkah 4. Anda dapat melakukan ini, misalnya, dengan mengubah Nama Unit Organisasi ke nama yang berbeda seperti yang ditunjukkan.

Masukkan atribut tambahan berikut untuk dikirim bersama permintaan sertifikat Anda.

- A challenge password [], misalnya: oracle123
- An optional company name [], misalnya: aws

12. Dapatkan tanda tangan sertifikat.

```
openssl req -noout -text -in certdms.csr | grep -i signature
```

Kunci tanda tangan untuk posting ini adalah sha256WithRSAEncryption.

13. Jalankan perintah berikut untuk menghasilkan file certificate (.crt).

```
openssl x509 -req -in certdms.csr -CA self-rootCA.pem -CAkey self-rootCA.key  
-CAcreateserial -out certdms.crt -days 365 -sha256
```

Ini menampilkan output seperti berikut ini.

```
Signature ok  
subject=/C=AU/ST=NSW/L=Sydney/O=awsweb/OU=DBeng/CN=aws  
Getting CA Private Key
```

14. Tambahkan sertifikat ke wallet.

```
orapki wallet add -wallet /u01/app/oracle/wallet -pwd oracle123 -user_cert -cert  
certdms.crt
```

15. Lihat dompetnya. Itu harus memiliki dua entri. Lihat kode berikut.

```
orapki wallet display -wallet /u01/app/oracle/wallet -pwd oracle123
```

16. Konfigurasi file 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. Hentikan Oracle listener.

```
lsnrctl stop
```

18. Tambahkan entri untuk SSL di file 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. Konfigurasi file `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. Nyalakan kembali Oracle listener.

```
lsnrctl start
```

21. Tampilkan status Oracle listener.

```
lsnrctl status
```

22. Uji sambungan SSL ke basis data dari localhost menggunakan sqlplus dan entri tnsnames SSL.

```
sqlplus -L ORACLE_USER@SID_ssl
```

23. Verifikasi bahwa Anda berhasil terhubung menggunakan SSL.

```
SELECT SYS_CONTEXT('USERENV', 'network_protocol') FROM DUAL;

SYS_CONTEXT('USERENV', 'NETWORK_PROTOCOL')
-----
tcps
```

24. Ubah direktori ke direktori dengan sertifikat yang ditandatangani sendiri.

```
cd /u01/app/oracle/self_signed_cert
```

25. Buat dompet Oracle klien baru AWS DMS untuk digunakan.

```
orapki wallet create -wallet ./ -auto_login_only
```

26. Tambahkan sertifikat root yang ditandatangani sendiri ke Oracle wallet.

```
orapki wallet add -wallet ./ -trusted_cert -cert self-rootCA.pem -auto_login_only
```

27. Buat daftar isi dompet Oracle AWS DMS untuk digunakan. Daftar harus mencakup sertifikat root yang ditandatangani sendiri.

```
orapki wallet display -wallet ./
```

Ini memiliki output seperti berikut ini.

```
Trusted Certificates:
Subject:          CN=aws,OU=DBeng,O=AmazonWebService,L=Sydney,ST=NSW,C=AU
```

28. Unggah dompet Oracle yang baru saja Anda buat. AWS DMS

Metode enkripsi yang didukung untuk menggunakan Oracle sebagai sumber AWS DMS

Dalam tabel berikut, Anda dapat menemukan metode enkripsi data transparan (TDE) yang AWS DMS mendukung saat bekerja dengan database sumber Oracle.

Metode akses log redo	Tablespace TDE	Kolom TDE
Oracle LogMiner	Ya	Ya
Binary Reader	Ya	Ya

AWS DMS mendukung Oracle TDE saat menggunakan Binary Reader, pada tingkat kolom dan tingkat tablespace. Untuk menggunakan enkripsi TDE AWS DMS, pertama-tama identifikasi lokasi dompet Oracle tempat kunci enkripsi TDE dan kata sandi TDE disimpan. Kemudian identifikasi kunci enkripsi TDE yang benar dan kata sandi untuk titik akhir sumber Oracle Anda.

Mengidentifikasi dan menentukan kunci enkripsi dan kata sandi untuk enkripsi TDE

1. Jalankan kueri berikut untuk menemukan enkripsi Oracle wallet pada host basis data Oracle.

```
SQL> SELECT WRL_PARAMETER FROM V$ENCRYPTION_WALLET;
```

```
WRL_PARAMETER
```

```
-----
/u01/oracle/product/12.2.0/dbhome_1/data/wallet/
```

Di sini, /u01/oracle/product/12.2.0/dbhome_1/data/wallet/ adalah lokasi wallet.

2. Dapatkan ID kunci utama menggunakan salah satu opsi enkripsi berikut, tergantung pada mana yang mengembalikan nilai ini.

a. Untuk tabel atau enkripsi tingkat kolom, jalankan kueri berikut.

```
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
-----
AWGDC9g1Sk8Xv+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Di sini, AWGDC9g1Sk8Xv+3bVveiVSg adalah ID kunci utama (MKEYID). Jika Anda mendapatkan nilai untuk MKEYID, Anda dapat melanjutkan dengan Langkah 3. Jika tidak, lanjutkan dengan Langkah 2.2.

 Note

Karakter string trailing 'A' (AAA . . .) bukan bagian dari nilai.

b. Untuk enkripsi tingkat tablespace, jalankan kueri berikut.

```
SQL> SELECT TABLESPACE_NAME, ENCRYPTED FROM dba_tablespaces;
TABLESPACE_NAME          ENC
-----
SYSTEM                   NO
SYSAUX                   NO
UNDOTBS1                 NO
TEMP                     NO
USERS                    NO
TEST_ENCRYPT              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$tablespace t, x$kcbbk x
WHERE t.ts#=x.ts#)
WHERE name = 'TEST_ENCRYPT';

NAME                                MASTERKEYID_BASE64
-----                                -
TEST_ENCRYPT                          AWGDC9g1Sk8Xv+3bVveiVSg=

```

Di sini, AWGDC9g1Sk8Xv+3bVveiVSg adalah ID kunci utama (TEST_ENCRYPT). Jika langkah 2.1 dan 2.2 mengembalikan nilai, maka selalu identik.

Karakter trailing '=' bukan bagian dari nilai.

3. Dari baris perintah, cantumkan entri wallet enkripsi pada host basis data Oracle sumber.

```

$ mkstore -wrl /u01/oracle/product/12.2.0/dbhome_1/data/wallet/ -list
Oracle Secret Store entries:
ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9g1Sk8Xv+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.DB.ENCRYPTION.AY1mRA80XU9Qvzo3idU40H4AAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.DB.ENCRYPTION.MASTERKEY
ORACLE.SECURITY.ID.ENCRYPTION.
ORACLE.SECURITY.KB.ENCRYPTION.
ORACLE.SECURITY.KM.ENCRYPTION.AY1mRA80XU9Qvzo3idU40H4AAAAAAAAAAAAAAAAAAAAAAAAAAAA

```

Temukan entri yang berisi ID kunci utama yang Anda temukan di langkah 2 (AWGDC9g1Sk8Xv+3bVveiVSg). Entri ini adalah nama kunci enkripsi TDE.

4. Lihat detail entri yang Anda temukan pada langkah sebelumnya.

```

$ mkstore -wrl /u01/oracle/product/12.2.0/dbhome_1/data/wallet/ -viewEntry
ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9g1Sk8Xv+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.AWGDC9g1Sk8Xv+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
= AEMAASAASGys0phWHfNt9J5mEMkkegGFid4LLfQszDojgDzbfoYDEACv0x3pJC+UGD/
PdtE2jLIcBQcAeHgJChQGLA==

```

Masukkan kata sandi wallet untuk melihat hasilnya.

Di sini, nilai di sebelah kanan '=' adalah kata sandi TDE.

5. Tentukan nama kunci enkripsi TDE untuk titik akhir sumber Oracle dengan menetapkan atribut sambungan tambahan `securityDbEncryptionName`.

```
securityDbEncryptionName=ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9g1Sk8Xv
+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

6. Berikan kata sandi TDE terkait untuk kunci ini pada konsol sebagai bagian dari nilai Kata sandi sumber Oracle. Gunakan urutan berikut untuk memformat nilai kata sandi yang dipisahkan dengan koma, diakhiri dengan nilai kata sandi TDE.

```
Oracle_db_password,ASM_Password,AEMAASAASGYs0phWHfNt9J5mEMkkegGFid4LLfQszDojgDzbfoYDEACv0x3pJC
+UGD/PdtE2jLIcBQcAeHgJChQGLA==
```

Tentukan nilai kata sandi dalam urutan ini terlepas dari konfigurasi basis data Oracle Anda. Misalnya, jika Anda menggunakan TDE tetapi basis data Oracle Anda tidak menggunakan ASM, tentukan nilai kata sandi dalam urutan yang dipisahkan koma berikut.

```
Oracle_db_password,,AEMAASAASGYs0phWHfNt9J5mEMkkegGFid4LLfQszDojgDzbfoYDEACv0x3pJC
+UGD/PdtE2jLIcBQcAeHgJChQGLA==
```

Jika kredensial TDE yang Anda tentukan salah, tugas AWS DMS migrasi tidak akan gagal. Namun, tugas tersebut juga tidak membaca atau menerapkan perubahan replikasi yang sedang berlangsung ke basis data target. Setelah memulai tugas, pantau Statistik tabel pada laman tugas migrasi konsol untuk memastikan bahwa perubahan telah direplikasi.

Jika DBA mengubah nilai kredensial TDE untuk basis data Oracle sementara tugas berjalan, tugas akan gagal. Pesan keliru berisi nama kunci enkripsi TDE baru. Untuk menentukan nilai baru dan memulai ulang tugas, gunakan prosedur sebelumnya.

Important

Anda tidak dapat memanipulasi dompet TDE yang dibuat di lokasi Oracle Automatic Storage Management (ASM) karena perintah tingkat OS seperti `cp`, `mv`, `rm`, dan `mkdir` merusak file dompet yang disimpan di lokasi ASM. Pembatasan ini khusus untuk file dompet TDE yang disimpan di lokasi ASM saja, tetapi tidak untuk file dompet TDE yang disimpan dalam direktori OS lokal.

Untuk memanipulasi dompet TDE yang disimpan di ASM dengan perintah tingkat OS, buat keystore lokal dan gabungkan keystore ASM ke keystore lokal sebagai berikut:

1. Buat keystore lokal.

```
ADMINISTER KEY MANAGEMENT create keystore file system wallet location
identified by wallet password;
```

2. Gabungkan keystore ASM ke keystore lokal.

```
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;
```

Kemudian, untuk membuat daftar entri dompet enkripsi dan kata sandi TDE, jalankan langkah 3 dan 4 terhadap keystore lokal.

Metode kompresi yang didukung untuk menggunakan Oracle sebagai sumber AWS DMS

Dalam tabel berikut, Anda dapat menemukan metode kompresi mana yang AWS DMS mendukung saat bekerja dengan database sumber Oracle. Seperti yang ditunjukkan tabel, dukungan kompresi bergantung pada versi database Oracle Anda dan apakah DMS dikonfigurasi untuk menggunakan Oracle LogMiner untuk mengakses log redo.

Versi	Basic	OLTP	HCC (dari Oracle 11g R2 atau versi lebih baru)	Lainnya
Oracle 10	Tidak	N/A	N/A	Tidak
Oracle 11 atau yang lebih baru - Oracle LogMiner	Ya	Ya	Ya	Ya - Metode kompresi apa pun yang

Versi	Basic	OLTP	HCC (dari Oracle 11g R2 atau versi lebih baru)	Lainnya
				didukung oleh Oracle LogMiner.
Oracle 11 atau versi yang lebih baru — Binary Reader	Ya	Ya	Ya — Untuk informasi lebih lanjut, lihat catatan berikut.	Ya

Note

Ketika titik akhir sumber Oracle dikonfigurasi untuk menggunakan Binary Reader, tingkat Query Low metode kompresi HCC didukung untuk tugas beban penuh saja.

Mereplikasi tabel bersarang menggunakan Oracle sebagai sumber AWS DMS

AWS DMS mendukung replikasi tabel Oracle yang berisi kolom yang merupakan tabel bersarang atau tipe yang ditentukan. Untuk mengaktifkan fungsionalitas ini, tambahkan pengaturan atribut sambungan tambahan berikut ke titik akhir sumber Oracle.

```
allowSelectNestedTables=true;
```

AWS DMS membuat tabel target dari tabel bersarang Oracle sebagai tabel induk dan anak biasa pada target tanpa kendala unik. Untuk mengakses data yang benar pada target, gabungkan tabel induk dan anak. Untuk melakukannya, pertama-tama secara manual buatlah indeks nonunik pada kolom NESTED_TABLE_ID dalam tabel anak target. Anda kemudian dapat menggunakan kolom NESTED_TABLE_ID di klausul join ON bersama-sama dengan kolom induk yang sesuai dengan nama tabel anak. Selain itu, membuat indeks seperti itu meningkatkan kinerja ketika data tabel anak target diperbarui atau dihapus oleh AWS DMS. Sebagai contoh, lihat [Contoh gabungan untuk tabel induk dan anak pada target](#).

Kami sarankan Anda mengonfigurasi tugas agar berhenti setelah beban penuh selesai. Kemudian, buat indeks nonunik ini untuk semua tabel anak yang telah direplikasi pada target dan lanjutkan tugas.

Jika tabel bersarang yang diambil ditambahkan ke tabel induk yang ada (ditangkap atau tidak ditangkap), tangani AWS DMS dengan benar. Namun, indeks nonunik untuk tabel target yang sesuai tidak dibuat. Dalam hal ini, jika tabel anak target menjadi sangat besar, performa mungkin akan terpengaruh. Dalam kasus seperti itu, kami merekomendasikan agar Anda menghentikan tugas, membuat indeks, kemudian melanjutkan tugas.

Setelah tabel bersarang direplikasi ke target, jalankan DBA untuk menggabungkan tabel induk dengan tabel anaknya yang sesuai untuk meratakan data.

Prasyarat untuk mereplikasi tabel bersarang Oracle sebagai sumber

Pastikan bahwa Anda mereplikasi tabel induk untuk semua tabel bersarang yang telah direplikasi. Sertakan tabel induk (tabel yang berisi kolom tabel bersarang) dan tabel turunan (yaitu, bersarang) dalam AWS DMS pemetaan tabel.

Jenis tabel bersarang yang didukung Oracle sebagai sumber

AWS DMS mendukung jenis tabel bersarang Oracle berikut sebagai sumber:

- Jenis data
- Objek yang ditetapkan pengguna

Keterbatasan dukungan AWS DMS untuk tabel bersarang Oracle sebagai sumber

AWS DMS memiliki batasan berikut dalam mendukung tabel bersarang Oracle sebagai sumber:

- AWS DMS hanya mendukung satu tingkat sarang meja.
- AWS DMS pemetaan tabel tidak memeriksa apakah tabel induk dan anak atau tabel dipilih untuk replikasi. Artinya, adalah memungkinkan untuk memilih tabel induk tanpa anak atau tabel anak tanpa induk.

Bagaimana AWS DMS mereplikasi tabel bersarang Oracle sebagai sumber

AWS DMS mereplikasi tabel induk dan bersarang ke target sebagai berikut:

- AWS DMS membuat tabel induk identik dengan sumbernya. DMS kemudian mendefinisikan kolom bersarang di induk sebagai RAW(16) dan termasuk acuan ke tabel bersarang induk di kolom NESTED_TABLE_IDnya.

- AWS DMS membuat tabel anak identik dengan sumber bersarang, tetapi dengan kolom tambahan bernama `NESTED_TABLE_ID`. Kolom ini memiliki jenis dan nilai yang sama dengan kolom bersarang induk yang sesuai dan memiliki arti yang sama.

Contoh gabungan untuk tabel induk dan anak pada target

Untuk meratakan tabel induk, jalankan gabungan antara tabel induk dan anak, seperti yang ditunjukkan dalam contoh berikut:

1. Buat Type tabel.

```
CREATE OR REPLACE TYPE NESTED_TEST_T AS TABLE OF VARCHAR(50);
```

2. Buat tabel induk dengan kolom jenis `NESTED_TEST_T` sebagaimana didefinisikan sebelumnya.

```
CREATE TABLE NESTED_PARENT_TEST (ID NUMBER(10,0) PRIMARY KEY, NAME NESTED_TEST_T)
NESTED TABLE NAME STORE AS NAME_KEY;
```

3. Ratakan tabel `NESTED_PARENT_TEST` menggunakan gabungan dengan tabel anak `NAME_KEY` di mana `CHILD.NESTED_TABLE_ID` cocok dengan `PARENT.NAME`.

```
SELECT ... FROM NESTED_PARENT_TEST PARENT, NAME_KEY CHILD WHERE CHILD.NESTED_
TABLE_ID = PARENT.NAME;
```

Menyimpan REDO di Oracle ASM saat menggunakan Oracle sebagai sumber AWS DMS

Untuk sumber Oracle dengan generasi REDO tinggi, menyimpan REDO di Oracle ASM dapat menguntungkan kinerja, terutama dalam konfigurasi RAC karena Anda dapat mengonfigurasi DMS untuk mendistribusikan pembacaan ASM REDO di semua node ASM.

Untuk memanfaatkan konfigurasi ini, gunakan atribut `asmServer` koneksi. Misalnya, string koneksi berikut mendistribusikan pembacaan DMS REDO di 3 node 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))
```

Saat menggunakan NFS untuk menyimpan Oracle REDO, penting untuk memastikan bahwa patch klien DNFS (NFS langsung) yang berlaku diterapkan, khususnya patch apa pun yang menangani bug Oracle 25224242. Untuk informasi tambahan, tinjau Publikasi Oracle berikut mengenai patch terkait klien NFS Langsung, Patch yang [Direkomendasikan untuk](#) Klien NFS Langsung.

Selain itu, untuk meningkatkan kinerja baca NFS, kami sarankan Anda meningkatkan nilai `rsize` dan `wsizesize` masuk `fstab` untuk volume NFS, seperti yang ditunjukkan pada contoh berikut.

```
NAS_name_here:/ora_DATA1_archive /u09/oradata/DATA1 nfs
rw,bg,hard,nointr,tcp,nfsvers=3,_netdev,
timeo=600,rsize=262144,wsizesize=262144
```

Juga, sesuaikan `tcp-max-xfer-size` nilainya sebagai berikut:

```
vserver nfs modify -vserver vserver -tcp-max-xfer-size 262144
```

Pengaturan titik akhir saat menggunakan Oracle sebagai sumber AWS DMS

Anda dapat menggunakan pengaturan endpoint untuk mengkonfigurasi database sumber Oracle Anda mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir sumber menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--oracle-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan endpoint yang dapat Anda gunakan dengan Oracle sebagai sumber.

Nama	Penjelasan
<code>AccessAlternateDirectly</code>	Atur atribut ini menjadi salah untuk menggunakan Binary Reader untuk menangkap perubahan data untuk Amazon RDS for Oracle sebagai sumber. Ini memberitahu instans DMS untuk tidak mengakses log redo melalui penggantian prefiks jalur tertentu menggunakan akses file langsung. Untuk informasi selengkapnya, lihat Mengkonfigurasi tugas CDC untuk menggunakan Binary Reader dengan RDS untuk sumber Oracle AWS DMS .

Nama	Penjelasan
	<p>Nilai default: benar</p> <p>Nilai yang benar: benar/salah</p> <p>Contoh: <code>--oracle-settings '{"AccessAlternateDirectly": false}'</code></p>
AdditionalArchivedLogDestId	<p>Atur atribut ini dengan <code>ArchivedLogDestId</code> pada pengaturan primer-standby. Atribut ini berguna dalam peralihan ketika basis data Oracle Data Guard digunakan sebagai sumber. Dalam hal ini, AWS DMS perlu mengetahui tujuan mana untuk mendapatkan log redo arsip untuk membaca perubahan. Ini karena primer sebelumnya saat ini adalah instans Standby setelah peralihan.</p> <p>Meskipun AWS DMS mendukung penggunaan <code>RESETLOGS</code> opsi Oracle untuk membuka database, jangan pernah gunakan <code>RESETLOGS</code> kecuali diperlukan. Untuk informasi tambahan tentang <code>RESETLOGS</code>, lihat Konsep Perbaikan Data RMAN dalam Panduan Pengguna Backup dan Pemulihan Basis Data Oracle®.</p> <p>Nilai yang valid: Arsipkan Id tujuan</p> <p>Contoh: <code>--oracle-settings '{"AdditionalArchivedLogDestId": 2}'</code></p>

Nama	Penjelasan
AddSupplementalLogging	<p>Atur atribut ini untuk mengatur supplemental logging tingkat tabel untuk basis data Oracle. Atribut ini memungkinkan salah satu dari berikut ini pada semua tabel yang dipilih untuk tugas migrasi, tergantung pada metadata tabel:</p> <ul style="list-style-type: none">• KOLOM KUNCI UTAMA Pencatatan tambahan• KOLOM KUNCI UNIK logging tambahan• SEMUA KOLOM logging tambahan <p>Nilai default: SALAH</p> <p>Nilai yang benar: benar/salah</p> <p>Contoh: <code>--oracle-settings '{"AddSupplementalLogging": false}'</code></p> <div data-bbox="461 957 1508 1226" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Jika Anda menggunakan opsi ini, Anda masih perlu mengaktifkan supplemental logging tingkat basis data seperti yang dibahas sebelumnya.</p></div>
AllowSelectNestedTables	<p>Atur atribut ini menjadi benar untuk mengaktifkan replikasi tabel Oracle yang berisi kolom yang merupakan tabel bersarang atau jenis yang telah didefinisikan. Untuk informasi selengkapnya, lihat Mereplikasi tabel bersarang menggunakan Oracle sebagai sumber AWS DMS.</p> <p>Nilai default: salah</p> <p>Nilai yang benar: benar/salah</p> <p>Contoh: <code>--oracle-settings '{"AllowSelectNestedTables": true}'</code></p>

Nama	Penjelasan
ArchivedLogDestId	<p>Menentukan ID tujuan untuk log redo diarsipkan. Nilai ini harus sama dengan angka di kolom <code>dest_id</code> dari tampilan <code>v\$archived_log</code>. Jika Anda bekerja dengan tujuan redo log tambahan, kami sarankan Anda menggunakan <code>AdditionalArchivedLogDestId</code> atribut untuk menentukan ID tujuan tambahan. Melakukan hal ini meningkatkan performa dengan memastikan bahwa log yang benar diakses dari awal.</p> <p>Nilai default: 1</p> <p>Nilai yang benar: Angka</p> <p>Contoh: <code>--oracle-settings '{"ArchivedLogDestId": 1}'</code></p>
ArchivedLogsOnly	<p>Saat bidang ini disetel ke Y, AWS DMS hanya mengakses log redo yang diarsipkan. Jika log redo yang diarsipkan disimpan di Oracle ASM saja, akun AWS DMS pengguna harus diberikan hak ASM.</p> <p>Nilai default: N</p> <p>Nilai yang benar: Y/N</p> <p>Contoh: <code>--oracle-settings '{"ArchivedLogsOnly": Y}'</code></p>

Nama	Penjelasan
asmUsePLSQLArray (Hanya ECA)	<p>Gunakan atribut koneksi tambahan (ECA) ini saat menangkap perubahan sumber dengan BinaryReader Pengaturan ini memungkinkan DMS untuk menyangga 50 pembacaan pada tingkat ASM per utas baca tunggal sambil mengontrol jumlah utas menggunakan atribut. <code>parallelASMReadThread</code> Ketika Anda mengatur atribut ini, pembaca AWS DMS biner menggunakan blok PL/SQL anonim untuk menangkap data redo dan mengirimkannya kembali ke contoh replikasi sebagai buffer besar. Ini mengurangi jumlah perjalanan pulang pergi ke sumbernya. Ini dapat secara signifikan meningkatkan kinerja pengambilan sumber, tetapi menghasilkan konsumsi memori PGA yang lebih tinggi pada Instans ASM. Masalah stabilitas mungkin muncul jika target memori tidak cukup. Anda dapat menggunakan rumus berikut untuk memperkirakan total penggunaan memori PGA instans ASM dengan satu tugas DMS:</p> $\text{number_of_redo_threads} * \text{parallelASMReadThreads} * 7 \text{ MB}$ <p>Nilai default: SALAH</p> <p>Nilai yang benar: benar/salah</p> <p>Contoh ECA: <code>asmUsePLSQLArray=true;</code></p>
ConvertTimestampWithZoneToUTC	<p>Tetapkan atribut ini <code>true</code> untuk mengonversi nilai stempel waktu kolom 'TIMESTAMP WITH TIME ZONE' dan 'TIMESTAMP WITH LOCAL TIME ZONE' ke UTC. Secara default nilai atribut ini adalah 'false' dan data akan direplikasi menggunakan zona waktu database sumber.</p> <p>Nilai default: SALAH</p> <p>Nilai yang benar: benar/salah</p> <p>Contoh: <code>--oracle-settings '{"ConvertTimestampWithZoneToUTC": true}'</code></p>

Nama	Penjelasan
EnableHomogenousPartitionOps	<p>Atur atribut ini <i>true</i> untuk mengaktifkan replikasi Oracle Partition dan SubPartition DDL Operations untuk migrasi Oracle Homogenous.</p> <p>Perhatikan bahwa fitur dan peningkatan ini diperkenalkan di AWS DMS versi 3.4.7.</p> <p>Nilai default: SALAH</p> <p>Nilai yang benar: benar/salah</p> <p>Contoh: <code>--oracle-settings '{"EnableHomogenousPartitionOps": true}'</code></p>
EnableHomogenousTablespace	<p>Mengatur atribut ini untuk mengaktifkan replikasi tablespace homogen dan membuat tabel atau indeks yang ada di bawah tablespace yang sama pada target.</p> <p>Nilai default: salah</p> <p>Nilai yang benar: benar/salah</p> <p>Contoh: <code>--oracle-settings '{"EnableHomogenousTablespace": true}'</code></p>

Nama	Penjelasan
EscapeCharacter	<p>Atur atribut ini ke karakter escape. Karakter escape ini memungkinkan Anda untuk membuat karakter wildcard tunggal berperilaku sebagai karakter normal dalam ekspresi pemetaan tabel. Untuk informasi selengkapnya, lihat Wildcard dalam pemetaan tabel.</p> <p>Nilai default: Null</p> <p>Nilai yang valid: Karakter apa pun selain karakter wildcard</p> <p>Contoh: <code>--oracle-settings '{"EscapeCharacter": "#"}'</code></p> <div data-bbox="461 684 1507 953" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Anda hanya dapat menggunakan <code>escapeCharacter</code> untuk nama tabel. Itu tidak luput karakter dari nama skema atau nama kolom.</p></div>
ExposeViews	<p>Gunakan atribut ini untuk menarik data sekali dari tampilan; Anda tidak dapat menggunakannya untuk replikasi yang sedang berlangsung. Ketika Anda mengekstraksi data dari tampilan, tampilan ditampilkan sebagai tabel pada skema target.</p> <p>Nilai default: salah</p> <p>Nilai yang benar: benar/salah</p> <p>Contoh: <code>--oracle-settings '{"ExposeViews": true}'</code></p>

Nama	Penjelasan
ExtraArch ivedLogDestIds	<p>Menentukan ID dari satu tujuan lagi untuk satu atau lebih log redo diarsipkan. ID ini adalah nilai kolom <code>dest_id</code> dalam tampilan <code>v\$archive_d_log</code>. Gunakan pengaturan ini dengan atribut koneksi <code>ArchivedLogDestId</code> tambahan dalam <code>primary-to-single</code> pengaturan atau <code>primary-to-multiple-standby</code> penyiapan.</p> <p>Pengaturan ini berguna dalam peralihan ketika Anda menggunakan database Oracle Data Guard sebagai sumber. Dalam hal ini, AWS DMS perlu informasi tentang tujuan apa untuk mendapatkan log redo arsip dari untuk membaca perubahan. AWS DMS membutuhkan ini karena setelah peralihan primer sebelumnya adalah instance siaga.</p> <p>Nilai yang valid: Arsipkan Id tujuan</p> <p>Contoh: <code>--oracle-settings '{"ExtraArchivedLogDestIds": 1}'</code></p>
FailTasks OnLobTrun cation	<p>Ketika diatur ke <code>true</code>, atribut ini menyebabkan tugas gagal jika ukuran sebenarnya dari kolom LOB lebih besar dari <code>LobMaxSize</code> yang ditentukan.</p> <p>Jika tugas diatur ke mode LOB terbatas dan opsi ini diatur menjadi <code>true</code>, tugas gagal dan bukannya momotong data LOB.</p> <p>Nilai default: salah</p> <p>Nilai yang benar: Boolean</p> <p>Contoh: <code>--oracle-settings '{"FailTasksOnLobTruncation": true}'</code></p>

Nama	Penjelasan
filterTransactionsOfUser (Hanya ECA)	<p>Gunakan atribut koneksi tambahan (ECA) ini untuk memungkinkan DMS mengabaikan transaksi dari pengguna tertentu saat mereplikasi data dari Oracle saat menggunakan. LogMiner Anda dapat meneruskan nilai nama pengguna yang dipisahkan koma, tetapi harus dalam semua huruf KAPITAL.</p> <p>Contoh ECA: <code>filterTransactionsOfUser= <i>USERNAME</i>;</code></p>
NumberDataTypeScale	<p>Menentukan skala angka. Anda dapat memilih menaikkan skala hingga 38, atau Anda dapat memilih -1 untuk FLOAT, atau -2 untuk VARCHAR. Secara default, jenis data ANGKA dikonversi ke presisi 38, skala 10.</p> <p>Nilai default: 10</p> <p>Values valid: -2 hingga 38 (-2 untuk VARCHAR, -1 untuk FLOAT)</p> <p>Contoh: <code>--oracle-settings '{"NumberDataTypeScale": 12}'</code></p> <div data-bbox="461 1037 1507 1451" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Pilih kombinasi skala presisi, -1 (FLOAT) atau -2 (VARCHAR)</p> <ul style="list-style-type: none"> . DMS mendukung kombinasi skala presisi yang didukung oleh Oracle. Jika presisi 39 atau lebih, pilih -2 (VARCHAR) . NumberDataTypeScale Pengaturan untuk database Oracle digunakan untuk tipe data NUMBER saja (tanpa presisi eksplisit dan definisi skala). </div>
OpenTransactionWindow	<p>Menyediakan jangka waktu dalam hitungan menit untuk memeriksa transaksi terbuka untuk tugas CDC saja.</p> <p>Nilai default: 0</p> <p>Nilai yang valid: Sebuah integer dari 0 sampai 240</p> <p>Contoh: <code>openTransactionWindow=15;</code></p>

Nama	Penjelasan
<p>OraclePathPrefix</p>	<p>Atur atribut string ini ke nilai yang diperlukan untuk menggunakan Binary Reader untuk menangkap perubahan data untuk Amazon RDS for Oracle sebagai sumber. Nilai ini menentukan root default Oracle yang digunakan untuk mengakses log redo. Untuk informasi selengkapnya, lihat Mengkonfigurasi tugas CDC untuk menggunakan Binary Reader dengan RDS untuk sumber Oracle AWS DMS.</p> <p>Nilai default: tidak ada.</p> <p>Nilai yang valid: /rdsdbdata/db/ORCL_A/</p> <p>Contoh: <code>--oracle-settings '{"OraclePathPrefix": " /rdsdbdata/db/ORCL_A/"}'</code></p>
<p>ParallelASMReadThreads</p>	<p>Atur atribut ini untuk mengubah jumlah thread yang dikonfigurasi DMS untuk melakukan change data capture (CDC) menggunakan Oracle Automatic Storage Management (ASM). Anda dapat menentukan nilai integer antara 2 (default) dan 8 (maksimum). Gunakan atribut ini bersama-sama dengan ReadAheadBlocks atribut. Untuk informasi selengkapnya, lihat Mengkonfigurasi tugas CDC untuk menggunakan Binary Reader dengan RDS untuk sumber Oracle AWS DMS.</p> <p>Nilai default: 2</p> <p>Nilai yang valid: Integer dari 2 sampai 8</p> <p>Contoh: <code>--oracle-settings '{"ParallelASMReadThreads": 6;}'</code></p>

Nama	Penjelasan
ReadAheadBlocks	<p>Atur atribut ini untuk mengubah jumlah blok baca-depan yang dikonfigurasi DMS untuk melakukan CDC menggunakan Oracle Automatic Storage Management (ASM) dan penyimpanan NAS non-ASM. Anda dapat menentukan nilai integer antara 1000 (default) dan 200.000 (maksimum). Gunakan atribut ini bersama dengan atribut <code>ParallelASMReadThreads</code>. Untuk informasi selengkapnya, lihat Mengkonfigurasi tugas CDC untuk menggunakan Binary Reader dengan RDS untuk sumber Oracle AWS DMS.</p> <p>Nilai default: 1000</p> <p>Nilai yang valid: Integer dari 1000 sampai 200.000</p> <p>Contoh: <code>--oracle-settings '{"ReadAheadBlocks": 150000}'</code></p>
ReadTableSpaceName	<p>Ketika diatur ke <code>true</code>, atribut ini mendukung replikasi tablespace.</p> <p>Nilai default: salah</p> <p>Nilai yang benar: Boolean</p> <p>Contoh: <code>--oracle-settings '{"ReadTableSpaceName": true}'</code></p>

Nama	Penjelasan
ReplacePathPrefix	<p>Atur atribut ini menjadi true untuk menggunakan Binary Reader untuk menangkap perubahan data untuk Amazon RDS for Oracle sebagai sumber. Pengaturan ini memberitahu instans DMS untuk menggantikan root Oracle default dengan pengaturan UsePathPrefix yang ditentukan untuk mengakses log redo. Untuk informasi selengkapnya, lihat Mengkonfigurasi tugas CDC untuk menggunakan Binary Reader dengan RDS untuk sumber Oracle AWS DMS.</p> <p>Nilai default: salah</p> <p>Nilai yang benar: benar/salah</p> <p>Contoh: <code>--oracle-settings '{"ReplacePathPrefix": true}'</code></p>
RetryInterval	<p>Menentukan jumlah detik bahwa sistem menunggu sebelum mengirim ulang kueri.</p> <p>Nilai default: 5</p> <p>Nilai yang valid: Angka mulai dari 1</p> <p>Contoh: <code>--oracle-settings '{"RetryInterval": 6}'</code></p>
SecurityDbEncryptionName	<p>Menentukan nama kunci yang digunakan untuk enkripsi data transparan (TDE) dari kolom dan tablespace dalam basis data sumber Oracle. Untuk informasi lebih lanjut tentang pengaturan atribut ini dan kata sandi terkait pada titik akhir sumber Oracle, lihat Metode enkripsi yang didukung untuk menggunakan Oracle sebagai sumber AWS DMS.</p> <p>Nilai default: ""</p> <p>Nilai valid: String</p> <p>Contoh: <code>--oracle-settings '{"SecurityDbEncryptionName": "ORACLE.SECURITY.DB.ENCRYPTION.Adg8m2dhkU/0v/m5QUaaNJEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}'</code></p>

Nama	Penjelasan
<code>SpatialSdo2GeoJsonFunctionName</code>	<p>Untuk Oracle versi 12.1 atau sumber versi sebelumnya yang bermigrasi ke target PostgreSQL, gunakan atribut ini untuk mengonversi SDO_GEOMETRY ke format GEOJSON.</p> <p>Secara default, AWS DMS memanggil fungsi <code>SDO2GEOJSON</code> kustom yang harus ada dan dapat diakses oleh AWS DMS pengguna. Atau Anda dapat membuat fungsi kustom Anda sendiri yang meniru operasi <code>SDO2GEOJSON</code> dan atur <code>SpatialSdo2GeoJsonFunctionName</code> untuk memanggilnya sebagai gantinya.</p> <p>Nilai default: <code>SDO2GEOJSON</code></p> <p>Nilai yang benar: String</p> <p>Contoh: <code>--oracle-settings '{"SpatialSdo2GeoJsonFunctionName": "myCustomSDO2GEOJSONFunction"}'</code></p>

Nama	Penjelasan
StandbyDelayTime	<p>Gunakan atribut ini untuk menentukan waktu dalam menit untuk penundaan sinkronisasi standby. Jika sumber merupakan basis data standby Active Data Guard, gunakan atribut ini untuk menentukan jeda waktu antara basis data primer dan standby.</p> <p>Di AWS DMS, Anda dapat membuat tugas Oracle CDC yang menggunakan instance siaga Active Data Guard sebagai sumber untuk mereplikasi perubahan yang sedang berlangsung. Melakukan hal ini menghilangkan kebutuhan untuk terhubung ke basis data aktif yang mungkin sedang dalam produksi.</p> <p>Nilai default: 0</p> <p>Nilai yang benar: Angka</p> <p>Contoh: <code>--oracle-settings '{"StandbyDelayTime": 1}'</code></p> <p>Catatan: Saat menggunakan DMS 3.4.6, 3.4.7 dan yang lebih tinggi, penggunaan pengaturan koneksi ini adalah opsional. Dalam versi terbaru dari DMS 3.4.6 dan versi 3.4.7, <i>dms_user</i> harus memiliki <code>select</code> izin pada <code>V_\$DATAGUARD_STATS</code> , memungkinkan DMS untuk menghitung waktu tunda siaga.</p>
UseAlternateFolderForOnline	<p>Atur atribut ini menjadi benar untuk menggunakan Binary Reader untuk menangkap perubahan data untuk Amazon RDS for Oracle sebagai sumber. Ini memberitahu instans DMS untuk menggunakan pengganti prefiks tertentu untuk mengakses semua log redo online. Untuk informasi selengkapnya, lihat Mengkonfigurasi tugas CDC untuk menggunakan Binary Reader dengan RDS untuk sumber Oracle AWS DMS.</p> <p>Nilai default: salah</p> <p>Nilai yang benar: benar/salah</p> <p>Contoh: <code>--oracle-settings '{"UseAlternateFolderForOnline": true}'</code></p>

Nama	Penjelasan
UseBfile	<p>Atur atribut ini menjadi Y untuk menangkap perubahan data menggunakan utilitas Binary Reader. Atur UseLogminerReader menjadi N untuk mengatur atribut ini menjadi Y. Untuk menggunakan Binary Reader dengan Amazon RDS for Oracle sebagai sumber, Anda tetapkan atribut tambahan. Untuk informasi lebih lanjut tentang pengaturan ini dan menggunakan Oracle Automatic Storage Management (ASM), lihat Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC.</p> <p>Catatan: Saat mengatur nilai ini sebagai Extra Connection Attribute (ECA), nilai yang valid adalah 'Y' dan 'N'. Saat menyetel nilai ini sebagai pengaturan titik akhir, nilai yang valid adalah true dan false.</p> <p>Nilai default: N</p> <p>Nilai yang valid: Y/N (saat menyetel nilai ini sebagai ECA); benar/salah (saat menyetel nilai ini sebagai pengaturan titik akhir).</p> <p>Contoh: <code>--oracle-settings '{"UseBfile": Y}'</code></p>

Nama	Penjelasan
UseLogminerReader	<p>Atur atribut ini ke Y untuk menangkap data perubahan menggunakan LogMiner utilitas (default). Atur opsi ini menjadi N jika Anda ingin AWS DMS untuk mengakses log redo sebagai file biner. Ketika Anda mengatur opsi ini menjadi N, tambahkan juga pengaturan UseFile=y. Untuk informasi lebih lanjut tentang pengaturan ini dan penggunaan Oracle Automatic Storage Management (ASM), lihat Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC.</p> <p>Catatan: Saat mengatur nilai ini sebagai Extra Connection Attribute (ECA), nilai yang valid adalah 'Y' dan 'N'. Saat menyetel nilai ini sebagai pengaturan titik akhir, nilai yang valid adalah true dan false.</p> <p>Nilai default: Y</p> <p>Nilai yang valid: Y/N (saat menyetel nilai ini sebagai ECA); benar/salah (saat menyetel nilai ini sebagai pengaturan titik akhir).</p> <p>Contoh: <code>--oracle-settings '{"UseLogminerReader": Y}'</code></p>
UsePathPrefix	<p>Atur atribut string ini ke nilai yang diperlukan untuk menggunakan Binary Reader untuk menangkap perubahan data untuk Amazon RDS for Oracle sebagai sumber. Nilai ini menentukan prefiks jalur yang digunakan untuk menggantikan root Oracle default untuk mengakses log redo. Untuk informasi selengkapnya, lihat Mengkonfigurasi tugas CDC untuk menggunakan Binary Reader dengan RDS untuk sumber Oracle AWS DMS.</p> <p>Nilai default: tidak ada.</p> <p>Nilai yang valid: <code>/rdsdbdata/log/</code></p> <p>Contoh: <code>--oracle-settings '{"UsePathPrefix": " /rdsdbdata/log/ "'}</code></p>

Jenis data sumber untuk Oracle

Endpoint Oracle untuk AWS DMS mendukung sebagian besar tipe data Oracle. Tabel berikut menunjukkan tipe data sumber Oracle yang didukung saat menggunakan AWS DMS dan pemetaan default ke tipe AWS DMS data.

Note

Dengan pengecualian tipe data LONG dan LONG RAW, ketika mereplikasi dari sumber Oracle ke target Oracle (replikasi homogen), semua tipe data sumber dan target akan identik. Tetapi tipe data LONG akan dipetakan ke CLOB dan tipe data LONG RAW akan dipetakan ke BLOB.

Untuk informasi tentang cara melihat jenis data yang dipetakan dalam target, lihat bagian titik akhir target yang Anda gunakan.

Untuk informasi tambahan tentang tipe AWS DMS data, lihat [Tipe data untuk AWS Database Migration Service](#).

Tipe data Oracle	AWS DMS tipe data
BINARY_FLOAT	REAL4
BINARY_DOUBLE	REAL8
BINARY	BYTES
FLOAT (P)	Jika presisi kurang dari atau sama dengan 24, gunakan REAL4. Jika presisi lebih besar dari 24, gunakan REAL8.
ANGKA (P,S)	Ketika skala lebih besar dari 0, gunakan NUMERIK. Ketika skala adalah 0: <ul style="list-style-type: none"> • Dan presisi kurang dari atau sama dengan 2, gunakan INT1. • Dan presisi lebih besar dari 2 dan kurang dari atau sama dengan 4, gunakan INT2.

Tipe data Oracle	AWS DMS tipe data
	<ul style="list-style-type: none"> • Dan presisi lebih besar dari 4 dan kurang dari atau sama dengan 9, gunakan INT4. • Dan presisi lebih besar dari 9, menggunakan NUMERIC. • Dan presisi lebih dari atau sama dengan skala, gunakan NUMERIC. <p>Ketika skala kurang dari 0, gunakan REAL8.</p>
DATE	DATETIME
INTERVAL_YEAR TO MONTH	STRING (dengan indikasi interval year_to_month)
INTERVAL_DAY TO SECOND	STRING (dengan indikasi interval day_to_second)
TIMESTAMP	DATETIME
TIMESTAMP WITH TIME ZONE	STRING (dengan indikasi timestamp_with_timezone)
TIMESTAMP WITH LOCAL TIME ZONE	STRING (dengan indikasi timestamp_with_local_zona waktu)
CHAR	STRING
VARCHAR2	STRING
NCHAR	WSTRING
NVARCHAR2	WSTRING
RAW	BYTES
REAL	REAL8

Tipe data Oracle	AWS DMS tipe data
BLOB	BLOB Untuk menggunakan tipe data ini AWS DMS, Anda harus mengaktifkan penggunaan tipe data BLOB untuk tugas tertentu. AWS DMS mendukung tipe data BLOB hanya dalam tabel yang menyertakan kunci utama.
CLOB	CLOB Untuk menggunakan tipe data ini AWS DMS, Anda harus mengaktifkan penggunaan tipe data CLOB untuk tugas tertentu. Selama CDC, AWS DMS mendukung tipe data CLOB hanya dalam tabel yang menyertakan kunci utama.
NCLOB	NCLOB Untuk menggunakan tipe data ini AWS DMS, Anda harus mengaktifkan penggunaan tipe data NCLOB untuk tugas tertentu. Selama CDC, AWS DMS mendukung tipe data NCLOB hanya dalam tabel yang menyertakan kunci utama.

Tipe data Oracle	AWS DMS tipe data
LONG	<p data-bbox="545 226 634 260">CLOB</p> <p data-bbox="545 306 1414 388">Tipe data LONG tidak didukung dalam mode penerapan yang dioptimalkan secara batch (mode TurboStream CDC).</p> <p data-bbox="545 434 1414 516">Untuk menggunakan tipe data ini dengan AWS DMS, aktifkan penggunaan LOB untuk tugas tertentu.</p> <p data-bbox="545 562 1474 644">Selama CDC atau beban penuh, AWS DMS mendukung tipe data LOB hanya dalam tabel yang memiliki kunci utama.</p> <p data-bbox="545 690 1507 1056">Juga, AWS DMS tidak mendukung mode LOB penuh untuk memuat kolom LONG. Sebagai gantinya, Anda dapat menggunakan mode LOB terbatas untuk memigrasikan kolom LONG ke target Oracle. Dalam mode LOB terbatas, AWS DMS potong data apa pun menjadi 64 KB yang Anda atur ke kolom LONG yang lebih panjang dari 64 KB. Untuk informasi selengkapnya tentang dukungan LOB di AWS DMS, lihat Mengatur dukungan LOB untuk database sumber dalam suatu tugas AWS DMS</p>

Tipe data Oracle	AWS DMS tipe data
LONG RAW	<p data-bbox="544 226 630 258">BLOB</p> <p data-bbox="544 306 1497 390">Tipe data LONG RAW tidak didukung dalam mode penerapan yang dioptimalkan secara batch (mode TurboStream CDC).</p> <p data-bbox="544 432 1414 516">Untuk menggunakan tipe data ini dengan AWS DMS, aktifkan penggunaan LOB untuk tugas tertentu.</p> <p data-bbox="544 558 1474 642">Selama CDC atau beban penuh, AWS DMS mendukung tipe data LOB hanya dalam tabel yang memiliki kunci utama.</p> <p data-bbox="544 684 1507 1056">Juga, AWS DMS tidak mendukung mode LOB penuh untuk memuat kolom LONG RAW. Sebagai gantinya, Anda dapat menggunakan mode LOB terbatas untuk memigrasikan kolom LONG RAW ke target Oracle. Dalam mode LOB terbatas, AWS DMS potong data apa pun menjadi 64 KB yang Anda atur ke kolom LONG RAW yang lebih panjang dari 64 KB. Untuk informasi selengkapnya tentang dukungan LOB di AWS DMS, lihat Mengatur dukungan LOB untuk database sumber dalam suatu tugas AWS DMS</p>
XMLTYPE	CLOB
SDO_GEOMETRY	<p data-bbox="544 1186 1101 1218">BLOB (ketika Oracle migrasi ke Oracle)</p> <p data-bbox="544 1260 1182 1291">CLOB (ketika Oracle migrasi ke PostgreSQL)</p>

Tabel Oracle yang digunakan sebagai sumber dengan kolom jenis data berikut tidak didukung dan tidak dapat direplikasi. Mereplikasi kolom dengan jenis data ini menghasilkan kolom nol.

- BFILE
- ROWID
- REF
- UROWID
- Jenis data yang ditentukan pengguna
- ANYDATA

- VARRAY

 Note

Kolom virtual tidak didukung.

Migrasi jenis data spasial Oracle

Data spasial mengidentifikasi informasi geometri untuk objek atau lokasi di suatu ruang. Dalam basis data Oracle, deskripsi geometris dari objek spasial disimpan dalam sebuah objek jenis SDO_GEOMETRY. Dalam objek ini, deskripsi geometris disimpan dalam satu baris dalam satu kolom dari tabel yang telah ditetapkan pengguna.

AWS DMS mendukung migrasi tipe Oracle SDO_GEOMETRY dari sumber Oracle ke target Oracle atau PostgreSQL.

Saat Anda memigrasikan tipe data spasial Oracle menggunakan AWS DMS, perhatikan pertimbangan ini:

- Ketika migrasi ke target Oracle, pastikan untuk secara manual mentransfer entri USER_SDO_GEOM_METADATA yang mencakup informasi tentang jenis.
- Saat bermigrasi dari titik akhir sumber Oracle ke titik akhir target PostgreSQL, buat kolom target. AWS DMS Kolom ini memiliki geometri default dan informasi jenis geografi dengan dimensi 2D dan spatial reference identifier (SRID) sama dengan nol (0). Contohnya GEOMETRY, 2, 0.
- Untuk Oracle versi 12.1 atau sumber versi sebelumnya yang migrasi ke target PostgreSQL, konversikan objek SDO_GEOMETRY menjadi format GEOJSON dengan menggunakan fungsi SDO2GEOJSON, atau atribut sambungan tambahan spatialSdo2GeoJsonFunctionName. Untuk informasi selengkapnya, lihat [Pengaturan titik akhir saat menggunakan Oracle sebagai sumber AWS DMS](#).
- AWS DMS mendukung migrasi Kolom Spasial Oracle hanya untuk mode LOB Penuh. AWS DMS tidak mendukung mode LOB Terbatas atau LOB Inline. Untuk informasi selengkapnya tentang mode LOB, lihat [Mengatur dukungan LOB untuk database sumber dalam suatu tugas AWS DMS](#).
- Karena AWS DMS hanya mendukung mode LOB Penuh untuk memigrasikan Kolom Spasial Oracle, tabel kolom memerlukan kunci utama dan kunci unik. Jika tabel tidak memiliki kunci utama dan kunci unik, tabel dilewati dari migrasi.

Menggunakan database Microsoft SQL Server sebagai sumber AWS DMS

Migrasikan data dari satu atau banyak database Microsoft SQL Server menggunakan AWS DMS. Dengan database SQL Server sebagai sumber, Anda dapat memigrasikan data ke database SQL Server lain, atau ke salah satu database lain AWS DMS yang didukung.

Untuk informasi tentang versi SQL Server yang AWS DMS mendukung sebagai sumber, lihat [Sumber untuk AWS DMS](#).

Basis data sumber SQL Server dapat dipasang pada setiap komputer dalam jaringan Anda. Akun SQL Server dengan hak istimewa akses yang sesuai untuk basis data sumber untuk jenis tugas yang Anda pilih diperlukan untuk digunakan dengan AWS DMS. Akun ini harus memiliki `view definition` dan `view server state` izin. Anda menambahkan izin ini menggunakan perintah berikut:

```
grant view definition to [user]
grant view server state to [user]
```

AWS DMS mendukung migrasi data dari contoh bernama SQL Server. Anda dapat menggunakan notasi berikut dalam nama server ketika Anda membuat titik akhir sumber.

```
IPAddress\InstanceName
```

Sebagai contoh, berikut ini adalah nama server titik akhir sumber yang benar. Di sini, bagian pertama dari nama adalah alamat IP server, dan bagian kedua adalah nama instans SQL Server (dalam contoh ini, SQLTest).

```
10.0.0.25\SQLTest
```

Juga, dapatkan nomor port yang instance bernama SQL Server Anda mendengarkan, dan menggunakannya untuk mengkonfigurasi titik akhir AWS DMS sumber Anda.

Note

Port 1433 adalah default untuk SQL Server. Tapi port dinamis yang berubah setiap kali SQL Server dimulai, dan nomor port statis tertentu yang digunakan untuk menyambung ke SQL Server melalui firewall juga sering digunakan. Jadi, Anda ingin mengetahui nomor port

sebenarnya dari instance bernama SQL Server saat Anda membuat titik akhir AWS DMS sumber.

Anda dapat menggunakan SSL untuk mengenkripsi sambungan antara titik akhir SQL Server dan instans replikasi. Untuk informasi selengkapnya tentang menggunakan titik akhir SSL dengan SQL Server, lihat [Menggunakan SSL dengan AWS Database Migration Service](#).

Untuk detail tambahan tentang bekerja dengan database sumber SQL Server dan AWS DMS, lihat berikut ini.

Topik

- [Keterbatasan dalam menggunakan SQL Server sebagai sumber AWS DMS](#)
- [Izin untuk tugas beban penuh saja](#)
- [Prasyarat untuk menggunakan replikasi berkelanjutan \(CDC\) dari sumber SQL Server](#)
- [Menangkap perubahan data untuk SQL Server yang dikelola sendiri di tempat atau di Amazon EC2](#)
- [Menyiapkan replikasi yang sedang berlangsung pada instance cloud SQL Server DB](#)
- [Pengaturan yang disarankan saat menggunakan Amazon RDS for SQL Server sebagai sumber untuk AWS DMS](#)
- [Metode kompresi yang didukung untuk SQL Server](#)
- [Bekerja dengan grup ketersediaan SQL Server AlwaysOn yang dikelola sendiri](#)
- [Persyaratan keamanan saat menggunakan SQL Server sebagai sumber AWS Database Migration Service](#)
- [Pengaturan titik akhir saat menggunakan SQL Server sebagai sumber AWS DMS](#)
- [Jenis sumber data untuk SQL Server](#)

Keterbatasan dalam menggunakan SQL Server sebagai sumber AWS DMS

Keterbatasan berikut berlaku saat menggunakan basis data SQL Server sebagai sumber untuk AWS DMS:

- Properti identitas untuk kolom tidak bermigrasi ke kolom basis data target.
- Titik akhir SQL Server tidak mendukung penggunaan tabel dengan kolom jarang.
- Autentikasi Windows tidak didukung.

- Perubahan bidang terhitung di SQL Server tidak direplikasi.
- Tabel temporal tidak didukung.
- Penggantian partisi SQL Server tidak didukung.
- Saat menggunakan utilitas WRITETEXT dan UPDATETEXT, AWS DMS tidak menangkap peristiwa yang diterapkan pada database sumber.
- Pola data manipulation language (DML) berikut ini tidak didukung.

```
SELECT * INTO new_table FROM existing_table
```

- Ketika menggunakan SQL Server sebagai sumber, enkripsi tingkat kolom tidak didukung.
- AWS DMS tidak mendukung audit tingkat server pada SQL Server 2008 atau SQL Server 2008 R2 sebagai sumber. Ini karena masalah yang diketahui dengan SQL Server 2008 dan 2008 R2. Misalnya, menjalankan perintah berikut AWS DMS menyebabkan kegagalan.

```
USE [master]
GO
ALTER SERVER AUDIT [my_audit_test-20140710] WITH (STATE=on)
GO
```

- Kolom geometri tidak didukung dalam mode lob penuh saat menggunakan SQL Server sebagai sumber. Sebaliknya, gunakan modus lob terbatas atau atur pengaturan tugas `InlineLobMaxSize` untuk menggunakan mode lob inline.
- Saat menggunakan database sumber Microsoft SQL Server dalam tugas replikasi, definisi SQL Server Replication Publisher tidak dihapus jika Anda menghapus tugas. Administrator sistem Microsoft SQL Server harus menghapus definisi tersebut dari Microsoft SQL Server.
- Migrasi data dari terikat skema dan non-schema-bound tampilan didukung untuk tugas yang hanya memuat penuh.
- Mengganti nama tabel menggunakan `sp_rename` tidak didukung (misalnya, `sp_rename 'Sales.SalesRegion', 'SalesReg';`)
- Mengganti nama kolom menggunakan `sp_rename` tidak didukung (misalnya, `sp_rename 'Sales.Sales.Region', 'RegID', 'COLUMN';`)
- AWS DMS tidak mendukung pemrosesan perubahan untuk mengatur dan menghapus nilai default kolom (menggunakan `ALTER COLUMN SET DEFAULT` klausa dengan `ALTER TABLE` pernyataan).
- AWS DMS tidak mendukung pemrosesan perubahan untuk mengatur nullabilitas kolom (menggunakan `ALTER COLUMN [SET|DROP] NOT NULL` klausa dengan pernyataan). `ALTER TABLE`

- Dengan SQL Server 2012 dan SQL Server 2014, saat menggunakan replikasi DMS dengan Availability Groups, basis data distribusi tidak dapat ditempatkan dalam grup ketersediaan. SQL 2016 mendukung penempatan database distribusi ke dalam grup ketersediaan, kecuali untuk database distribusi yang digunakan dalam topologi gabungan, dua arah, atau replikasi. peer-to-peer
- Untuk tabel yang dipartisi, AWS DMS tidak mendukung pengaturan kompresi data yang berbeda untuk setiap partisi.
- Saat memasukkan nilai ke jenis data spasial SQL Server (GEOGRAPHY dan GEOMETRY), Anda dapat mengabaikan properti spatial reference system identifier (SRID) atau menentukan angka yang berbeda. Saat mereplikasi tabel dengan tipe data spasial, AWS DMS ganti SRID dengan SRID default (0 untuk GEOMETRI dan 4326 untuk GEOGRAFI).
- Jika basis data Anda tidak dikonfigurasi untuk MS-REPLICATION atau MS-CDC, Anda masih dapat menangkap tabel yang tidak memiliki Kunci Primer, tetapi hanya peristiwa INSERT/DELETE DML yang ditangkap. Peristiwa UPDATE dan TRUNCATE TABLE diabaikan.
- Indeks Columnstore tidak didukung.
- Tabel yang memorinya dioptimalkan (menggunakan In-Memory OLTP) tidak didukung.
- Ketika mereplikasi tabel dengan kunci primer yang terdiri dari beberapa kolom, memperbarui kolom kunci primer selama beban penuh tidak didukung.
- Ketahanan tertunda tidak didukung.
- Pengaturan `readBackupOnly=Y` titik akhir (atribut koneksi tambahan) tidak berfungsi pada RDS untuk instance sumber SQL Server karena cara RDS melakukan pencadangan.
- `EXCLUSIVE_AUTOMATIC_TRUNCATION` tidak berfungsi pada instance sumber Amazon RDS SQL Server karena pengguna RDS tidak memiliki akses untuk menjalankan prosedur tersimpan SQL Server, `sp_repldone`
- AWS DMS tidak menangkap perintah memotong.
- AWS DMS tidak mendukung replikasi dari database dengan percepatan pemulihan database (ADR) diaktifkan.
- AWS DMS tidak mendukung pengambilan pernyataan bahasa definisi data (DDL) dan bahasa manipulasi data (DML/bahasa manipulasi data) dalam satu transaksi.
- AWS DMS tidak mendukung replikasi paket aplikasi tingkat data (DACPAC).
- Pernyataan UPDATE yang melibatkan kunci primer atau indeks unik dan memperbarui beberapa baris data, dapat menyebabkan konflik saat Anda menerapkan perubahan pada database target. Ini mungkin terjadi, misalnya, ketika database target menerapkan pembaruan sebagai

pernyataan INSERT dan DELETE, bukan pernyataan UPDATE tunggal. Dengan mode penerapan yang dioptimalkan batch, tabel mungkin diabaikan. Dengan mode penerapan transaksional, operasi UPDATE dapat mengakibatkan pelanggaran kendala. Untuk menghindari masalah ini, muat ulang tabel yang relevan. Atau, cari catatan bermasalah di tabel kontrol Apply Exceptions (`dmslogs.aws_dms_apply_exceptions`) dan edit secara manual di database target. Untuk informasi selengkapnya, lihat [Mengubah pengaturan penyetelan pemrosesan](#).

- AWS DMS tidak mendukung replikasi tabel dan skema, di mana nama menyertakan karakter khusus dari set berikut.

```
\\ -- \n \" \b \r ' \t ;
```

- Penyembunyian data tidak didukung. AWS DMS memigrasikan data bertopeng tanpa masking.
- AWS DMS mereplikasi hingga 32.767 tabel dengan kunci utama dan hingga 1.000 kolom untuk setiap tabel. Ini karena AWS DMS membuat artikel replikasi SQL Server untuk setiap tabel yang direplikasi, dan artikel replikasi SQL Server memiliki keterbatasan ini.
- Saat menggunakan Change Data Capture (CDC), Anda harus menentukan semua kolom yang membentuk indeks unik sebagai NOT NULL. Jika persyaratan ini tidak terpenuhi, kesalahan sistem SQL Server 22838 akan terjadi.

Keterbatasan berikut berlaku saat mengakses log transaksi backup:

- Backup terenkripsi tidak didukung.
- Backup yang disimpan di URL atau Windows Azure tidak didukung.
- AWS DMS tidak mendukung pemrosesan langsung cadangan log transaksi di tingkat file dari folder bersama alternatif.

Izin untuk tugas beban penuh saja

Izin berikut diperlukan untuk melakukan tugas beban penuh saja. Perhatikan bahwa AWS DMS tidak membuat `dms_user` login. Untuk informasi tentang membuat login untuk SQL Server, lihat [Membuat pengguna database dengan 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;
```

Prasyarat untuk menggunakan replikasi berkelanjutan (CDC) dari sumber SQL Server

Anda dapat menggunakan replikasi berkelanjutan (mengubah pengambilan data, atau CDC) untuk database SQL Server yang dikelola sendiri di tempat atau di Amazon EC2, atau database cloud seperti Amazon RDS atau instans yang dikelola Microsoft Azure SQL.

Persyaratan berikut berlaku secara khusus saat menggunakan replikasi yang sedang berlangsung dengan database SQL Server sebagai sumber untuk: AWS DMS

- SQL Server harus dikonfigurasi untuk backup penuh, dan Anda harus menjalankan backup sebelum mulai mereplikasi data.
- Model pemulihan harus diatur menjadi Bulk logged atau Full.
- Backup SQL Server untuk beberapa disk tidak didukung. Jika cadangan didefinisikan untuk menulis cadangan database ke beberapa file melalui disk yang berbeda, tidak AWS DMS dapat membaca data dan AWS DMS tugas gagal.
- Untuk sumber SQL Server yang dikelola sendiri, definisi SQL Server Replication Publisher untuk sumber yang digunakan dalam tugas DMS CDC tidak dihapus ketika Anda menghapus tugas. Administrator sistem SQL Server harus menghapus definisi ini dari SQL Server untuk sumber yang dikelola sendiri.
- Selama CDC, AWS DMS perlu mencari cadangan log transaksi SQL Server untuk membaca perubahan. AWS DMS tidak mendukung cadangan log transaksi SQL Server yang dibuat menggunakan perangkat lunak cadangan pihak ketiga yang tidak dalam format asli. Untuk mendukung backup log transaksi yang ada dalam format asli dan dibuat menggunakan perangkat lunak backup pihak ketiga, tambahkan atribut sambungan `use3rdPartyBackupDevice=Y` ke titik akhir sumber.
- Untuk sumber SQL Server yang dikelola sendiri, perhatikan bahwa SQL Server tidak menangkap perubahan pada tabel yang baru dibuat sampai mereka telah diterbitkan. Ketika tabel ditambahkan ke sumber SQL Server, AWS DMS mengelola pembuatan publikasi. Namun, proses ini mungkin memakan waktu beberapa menit. Operasi yang dibuat untuk tabel yang baru dibuat selama penundaan ini tidak ditangkap atau direplikasi ke target.
- AWS DMS mengubah pengambilan data memerlukan pencatatan transaksi penuh untuk dihidupkan di SQL Server. Untuk mengaktifkan login transaksi penuh di SQL Server, aktifkan MS-REPLICATION atau CHANGE DATA CAPTURE (CDC).

- Entri tlog SQL Server tidak akan ditandai untuk digunakan kembali sampai pekerjaan penangkapan MS CDC memproses perubahan tersebut.
- Operasi CDC tidak didukung pada tabel yang memorinya dioptimalkan. Batasan ini berlaku untuk SQL Server 2014 (ketika fitur pertama kali diperkenalkan) dan lebih tinggi.
- AWS DMS mengubah pengambilan data memerlukan database distribusi secara default di Amazon EC2 atau On-Prem SQL server sebagai sumber. Jadi, pastikan Anda telah mengaktifkan distributor saat mengonfigurasi replikasi MS untuk tabel dengan kunci utama.

Menangkap perubahan data untuk SQL Server yang dikelola sendiri di tempat atau di Amazon EC2

Untuk menangkap perubahan dari database Microsoft SQL Server sumber, pastikan database dikonfigurasi untuk backup penuh. Konfigurasikan basis data dalam modus pemulihan penuh atau mode bulk-logged.

Untuk sumber SQL Server yang dikelola sendiri, AWS DMS gunakan yang berikut ini:

MS-Replication

Untuk menangkap perubahan untuk tabel dengan kunci primer. Anda dapat mengonfigurasi ini secara otomatis dengan memberikan hak istimewa sysadmin kepada pengguna AWS DMS endpoint pada instance SQL Server sumber. Atau Anda dapat mengikuti langkah-langkah di bagian ini untuk menyiapkan sumber dan menggunakan pengguna yang tidak memiliki hak sysadmin untuk titik akhir. AWS DMS

MS-CDC

Untuk menangkap perubahan untuk tabel tanpa kunci primer. Aktifkan MS-CDC pada tingkat basis data dan untuk semua tabel secara masing-masing.

Saat menyiapkan basis data SQL Server untuk replikasi yang sedang berlangsung (CDC), Anda dapat melakukan salah satu dari berikut ini.

- Atur replikasi yang sedang berlangsung menggunakan peran sysadmin.
- Atur replikasi yang sedang berlangsung untuk tidak menggunakan peran sysadmin.

Menyiapkan replikasi yang sedang berlangsung pada SQL Server yang dikelola sendiri

Bagian ini berisi informasi tentang pengaturan replikasi yang sedang berlangsung pada server SQL yang dikelola sendiri dengan atau tanpa menggunakan peran sysadmin.

Topik

- [Menyiapkan replikasi yang sedang berlangsung pada SQL Server yang dikelola sendiri: Menggunakan peran sysadmin](#)
- [Menyiapkan replikasi yang sedang berlangsung pada SQL Server mandiri: Tanpa peran sysadmin](#)

Menyiapkan replikasi yang sedang berlangsung pada SQL Server yang dikelola sendiri: Menggunakan peran sysadmin

AWS DMS replikasi berkelanjutan untuk SQL Server menggunakan replikasi SQL Server asli untuk tabel dengan kunci utama, dan mengubah pengambilan data (CDC) untuk tabel tanpa kunci utama.

Sebelum menyiapkan replikasi yang sedang berlangsung, lihat [Prasyarat untuk menggunakan replikasi berkelanjutan \(CDC\) dari sumber SQL Server](#).

Untuk tabel dengan kunci utama, umumnya AWS DMS dapat mengkonfigurasi artefak yang diperlukan pada sumber. Namun, untuk instans sumber SQL Server yang dikelola sendiri, pastikan untuk terlebih dahulu mengonfigurasi distribusi SQL Server secara manual. Setelah Anda melakukannya, pengguna AWS DMS sumber dengan izin sysadmin dapat secara otomatis membuat publikasi untuk tabel dengan kunci utama.

Untuk memeriksa apakah distribusi sudah dikonfigurasi, jalankan perintah berikut.

```
sp_get_distributor
```

Jika hasilnya NULL untuk distribusi kolom, maka distribusi tidak dikonfigurasi. Anda dapat menggunakan prosedur berikut untuk mengatur distribusi.

Mengatur distribusi

1. Sambungkan ke basis data sumber SQL Server Anda menggunakan alat SQL Server Management Studio (SSMS).
2. Buka menu konteks (klik kanan) untuk folder Replikasi, dan pilih Konfigurasi Distribusi. Wizard Konfigurasi Distribusi muncul.

3. Ikuti wizard untuk memasukkan nilai default dan membuat distribusi.

Untuk mengatur CDC

AWS DMS versi 3.4.7 dan yang lebih besar dapat mengatur MS CDC untuk database Anda dan semua tabel Anda secara otomatis jika Anda tidak menggunakan replika hanya-baca. Untuk menggunakan fitur ini, atur `SetUpMsCdcForTables` ECA ke `true`. Untuk informasi tentang ECA, lihat [Pengaturan titik akhir](#).

Untuk versi yang AWS DMS lebih awal dari 3.4.7, atau untuk replika hanya-baca sebagai sumber, lakukan langkah-langkah berikut:

1. Untuk tabel tanpa kunci primer, atur MS-CDC untuk basis data. Untuk melakukannya, gunakan akun yang memiliki peran `sysadmin`, dan jalankan perintah berikut.

```
use [DBname]
EXEC sys.sp_cdc_enable_db
```

2. Selanjutnya, atur MS-CDC untuk setiap tabel sumber. Untuk setiap tabel dengan kunci unik tetapi tanpa kunci primer, jalankan kueri berikut untuk mengatur 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. Untuk setiap tabel tanpa bukti kunci primer atau tanpa kunci unik, jalankan kueri berikut untuk mengatur MS-CDC.

```
exec sys.sp_cdc_enable_table
@source_schema = N'schema_name',
@source_name = N'table_name',
@role_name = NULL
GO
```

Untuk informasi lebih lanjut tentang pengaturan MS-CDC untuk tabel tertentu, lihat [Dokumentasi SQL Server](#).

Menyiapkan replikasi yang sedang berlangsung pada SQL Server mandiri: Tanpa peran sysadmin

Untuk informasi tentang menyiapkan replikasi yang sedang berlangsung pada SQL Server mandiri tanpa peran sysadmin, lihat [Menyiapkan replikasi yang sedang berlangsung pada SQL Server mandiri: Tanpa peran sysadmin](#)

Menyiapkan replikasi yang sedang berlangsung pada instance cloud SQL Server DB

Bagian ini menjelaskan cara mengatur CDC pada instance database SQL Server yang dihosting cloud. Instance SQL server yang dihosting cloud adalah instance yang berjalan di Amazon RDS for SQL Server, Instans Azure SQL Manged, atau instans SQL Server cloud terkelola lainnya. Untuk informasi tentang batasan replikasi yang sedang berlangsung untuk setiap jenis database, lihat [Keterbatasan dalam menggunakan SQL Server sebagai sumber AWS DMS](#).

Sebelum menyiapkan replikasi yang sedang berlangsung, lihat [Prasyarat untuk menggunakan replikasi berkelanjutan \(CDC\) dari sumber SQL Server](#).

Tidak seperti sumber Microsoft SQL Server yang dikelola sendiri, Amazon RDS for SQL Server tidak mendukung replikasi MS. Oleh karena itu, AWS DMS perlu menggunakan MS-CDC untuk tabel dengan atau tanpa kunci primer.

Amazon RDS tidak memberikan hak sysadmin untuk menyetel artefak replikasi yang AWS DMS digunakan untuk perubahan berkelanjutan dalam instance SQL Server sumber. Pastikan untuk mengaktifkan MS-CDC untuk instans Amazon RDS (menggunakan hak pengguna master) seperti pada prosedur berikut.

Untuk mengaktifkan MS-CDC untuk instance cloud SQL Server DB

1. Jalankan salah satu query berikut di tingkat database.

Untuk contoh RDS untuk SQL Server DB, gunakan kueri ini.

```
exec msdb.dbo.rds_cdc_enable_db 'DB_name'
```

Untuk instans DB terkelola Azure SQL, gunakan kueri ini.

```
USE DB_name  
GO  
EXEC sys.sp_cdc_enable_db  
GO
```

2. Untuk setiap tabel dengan kunci utama, jalankan query berikut untuk mengaktifkan 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
```

Untuk setiap tabel dengan kunci unik tetapi tidak ada kunci utama, jalankan kueri berikut untuk mengaktifkan 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
```

Untuk setiap tabel tanpa kunci primer atau kunci unik, jalankan kueri berikut untuk mengaktifkan MS-CDC.

```
exec sys.sp_cdc_enable_table
@source_schema = N'schema_name',
@source_name = N'table_name',
@role_name = NULL
GO
```

3. Tetapkan periode retensi agar perubahan tersedia di sumber dengan perintah berikut.

```
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'
```

Parameter `@pollinginterval` diukur dalam hitungan detik dengan nilai yang disarankan, yang ditetapkan menjadi 86399. Ini berarti bahwa log transaksi mempertahankan perubahan selama 86.399 detik (satu hari) ketika `@pollinginterval = 86399`. Prosedur `exec sp_cdc_start_job 'capture'` memulai pengaturan.

Note

Dengan beberapa versi SQL Server, jika nilai `pollinginterval` diatur ke lebih dari 3599 detik, nilai reset ke default lima detik. Ketika ini terjadi, entri T-Log dibersihkan sebelum AWS DMS dapat membacanya. Untuk menentukan versi SQL Server mana yang terpengaruh oleh masalah yang diketahui ini, lihat [artikel Microsoft KB ini](#).

Jika Anda menggunakan Amazon RDS dengan Multi-AZ, pastikan Anda juga mengatur sekunder Anda untuk memiliki nilai yang tepat jika terjadi failover.

```
exec rdsadmin..rds_set_configuration 'cdc_capture_pollinginterval' , 86399
```

Jika tugas AWS DMS replikasi yang menangkap perubahan yang sedang berlangsung pada sumber SQL Server Anda berhenti selama lebih dari satu jam, gunakan prosedur berikut.

Untuk mempertahankan periode retensi selama tugas AWS DMS replikasi

1. Hentikan tugas yang memotong log transaksi dengan menggunakan perintah berikut.

```
exec sp_cdc_stop_job 'capture'
```

2. Temukan tugas Anda di AWS DMS konsol dan lanjutkan tugas.
3. Pilih tab Monitoring, dan periksa CDCLatencySource metriknya.
4. Setelah metrik CDCLatencySource sama dengan 0 (nol) dan tidak berubah, ulang kembali tugas yang memotong log transaksi menggunakan perintah berikut.

```
exec sp_cdc_start_job 'capture'
```

Jangan lupa untuk memulai tugas yang memotong log transaksi SQL Server. Jika tidak, penyimpanan pada instans SQL Server Anda dapat terisi.

Batasan untuk replikasi yang sedang berlangsung pada instance cloud SQL Server DB

- AWS DMS mendukung replikasi berkelanjutan (CDC) dengan log transaksi aktif saja. Anda tidak dapat menggunakan log cadangan dengan CDC.

- Anda mungkin kehilangan peristiwa jika Anda memindahkannya dari log transaksi aktif ke log cadangan, atau memotongnya dari log transaksi aktif.

Pengaturan yang disarankan saat menggunakan Amazon RDS for SQL Server sebagai sumber untuk AWS DMS

Saat Anda bekerja dengan Amazon RDS for SQL Server sebagai sumber, tugas pengambilan bergantung pada parameter dan. `maxscans` `maxtrans` Parameter ini mengatur jumlah maksimum pemindaian yang dilakukan penangkapan pada log transaksi dan jumlah transaksi yang diproses untuk setiap pemindaian.

Untuk database, di mana sejumlah transaksi lebih besar dari `maxtrans*maxscans`, meningkatkan `polling_interval` nilai dapat menyebabkan akumulasi catatan log transaksi aktif. Pada gilirannya, akumulasi ini dapat menyebabkan peningkatan ukuran log transaksi.

Perhatikan bahwa AWS DMS tidak bergantung pada pekerjaan penangkapan MS-CDC. Pekerjaan penangkapan MS-CDC menandai entri log transaksi sebagai telah diproses. Hal ini memungkinkan tugas backup log transaksi untuk menghapus entri dari log transaksi.

Kami menyarankan Anda memantau ukuran log transaksi dan keberhasilan pekerjaan MS-CDC. Jika pekerjaan MS-CDC gagal, log transaksi dapat tumbuh secara berlebihan dan menyebabkan kegagalan replikasi. AWS DMS Anda dapat memantau kesalahan pekerjaan pengambilan MS-CDC menggunakan tampilan manajemen `sys.dm_cdc_errors` dinamis di database sumber. Anda dapat memantau ukuran log transaksi menggunakan perintah `DBCC SQLPERF(LOGSPACE)` manajemen.

Untuk mengatasi peningkatan log transaksi yang disebabkan oleh MS-CDC

1. Periksa apakah database `Log Space Used %` AWS DMS mereplikasi dari dan memvalidasi bahwa itu meningkat terus menerus.

```
DBCC SQLPERF(LOGSPACE)
```

2. Identifikasi apa yang memblokir proses pencadangan log transaksi.

```
Select log_reuse_wait, log_reuse_wait_desc, name from sys.databases where name = db_name();
```

Jika `log_reuse_wait_desc` nilainya sama `REPLICATION`, retensi cadangan log disebabkan oleh latensi di MS-CDC.

3. Tingkatkan jumlah peristiwa yang diproses oleh pekerjaan penangkapan dengan meningkatkan nilai `maxtrans` dan `maxscans` parameter.

```
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'
```

Untuk mengatasi masalah ini, tetapkan nilai `maxscans` dan `maxtrans` sehingga `maxtrans*maxscans` sama dengan jumlah rata-rata peristiwa yang dihasilkan untuk tabel yang AWS DMS mereplikasi dari database sumber untuk setiap hari.

Jika Anda menetapkan parameter ini lebih tinggi dari nilai yang disarankan, pekerjaan penangkapan akan memproses semua peristiwa di log transaksi. Jika Anda menetapkan parameter ini di bawah nilai yang disarankan, latensi MS-CDC meningkat dan log transaksi Anda bertambah.

Mengidentifikasi nilai yang sesuai untuk `maxscans` dan `maxtrans` bisa sulit karena perubahan beban kerja menghasilkan berbagai jumlah peristiwa. Dalam hal ini, kami menyarankan Anda mengatur pemantauan pada latensi MS-CDC. Untuk informasi selengkapnya, lihat [Memantau proses](#) dalam dokumentasi SQL Server. Kemudian konfigurasi `maxtrans` dan `maxscans` secara dinamis berdasarkan hasil pemantauan.

Jika AWS DMS tugas tidak dapat menemukan nomor urutan log (LSNs) yang diperlukan untuk melanjutkan atau melanjutkan tugas, tugas mungkin gagal dan memerlukan pemuatan ulang lengkap.

Note

Saat menggunakan AWS DMS untuk mereplikasi data dari sumber RDS untuk SQL Server, Anda mungkin mengalami kesalahan saat mencoba melanjutkan replikasi setelah peristiwa stop-start dari instans Amazon RDS. Hal ini disebabkan proses SQL Server Agent memulai kembali proses pengambilan pekerjaan ketika restart setelah peristiwa stop-start. Ini melewati interval pemungutan suara MS-CDC.

Karena itu, pada database dengan volume transaksi lebih rendah dari pemrosesan pekerjaan penangkapan MS-CDC, ini dapat menyebabkan data diproses atau ditandai sebagai direplikasi dan dicadangkan sebelum AWS DMS dapat dilanjutkan dari tempat berhenti, yang mengakibatkan kesalahan berikut:

```
[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)
```

Untuk mengurangi masalah ini, tetapkan `maxscans` nilai `maxtrans` dan seperti yang direkomendasikan sebelumnya.

Metode kompresi yang didukung untuk SQL Server

Perhatikan hal berikut tentang dukungan untuk metode kompresi SQL Server di AWS DMS:

- AWS DMS mendukung kompresi Baris/Halaman di SQL Server versi 2008 dan yang lebih baru.
- AWS DMS tidak mendukung format penyimpanan Vardecimal.
- AWS DMS tidak mendukung kolom jarang dan kompresi struktur kolom.

Bekerja dengan grup ketersediaan SQL Server AlwaysOn yang dikelola sendiri

Grup ketersediaan SQL Server Always On menyediakan ketersediaan tinggi dan pemulihan bencana sebagai alternatif tingkat perusahaan untuk pencerminan basis data.

Di AWS DMS, Anda dapat memigrasikan perubahan dari satu replika grup ketersediaan primer atau sekunder.

Bekerja dengan replika grup ketersediaan utama

Untuk menggunakan grup ketersediaan utama sebagai sumber AWS DMS, lakukan hal berikut:

1. Aktifkan opsi distribusi untuk semua instance SQL Server di replika ketersediaan Anda. Untuk informasi selengkapnya, lihat [Menyiapkan replikasi yang sedang berlangsung pada SQL Server yang dikelola sendiri](#).
2. Di AWS DMS konsol, buka pengaturan database sumber SQL Server. Untuk Nama Server, tentukan nama Layanan Nama Domain (DNS) atau alamat IP yang dikonfigurasi untuk pendengar grup ketersediaan Anda.

Ketika Anda memulai AWS DMS tugas untuk pertama kalinya, mungkin perlu waktu lebih lama dari biasanya untuk memulai. Kelambatan ini terjadi karena pembuatan artikel tabel sedang digandakan oleh server grup ketersediaan.

Bekerja dengan replika grup ketersediaan sekunder

Untuk menggunakan grup ketersediaan sekunder sebagai sumber AWS DMS, lakukan hal berikut:

1. Gunakan kredensial yang sama untuk menghubungkan ke replika individual seperti yang digunakan oleh pengguna titik akhir AWS DMS sumber.
2. Pastikan instans AWS DMS replikasi Anda dapat menyelesaikan nama DNS untuk semua replika yang ada, dan menghubungkannya. Anda dapat menggunakan query SQL berikut untuk mendapatkan nama DNS untuk semua replika Anda.

```
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. Saat Anda membuat titik akhir sumber, tentukan nama DNS pendengar grup ketersediaan untuk nama Server titik akhir atau untuk alamat Server rahasia titik akhir. Untuk informasi selengkapnya tentang pendengar grup ketersediaan, lihat [Apa itu pendengar grup ketersediaan?](#) dalam dokumentasi SQL Server.

Anda dapat menggunakan server DNS publik atau server DNS lokal untuk menyelesaikan pemroses grup ketersediaan, replika utama, dan replika sekunder. Untuk menggunakan server DNS lokal, konfigurasi Resolver Amazon Route 53. Untuk informasi selengkapnya, lihat [Menggunakan server nama on-premise Anda sendiri](#).

4. Tambahkan atribut koneksi tambahan berikut ke titik akhir sumber Anda.

Atribut koneksi tambahan	Nilai	Catatan
applicationIntent	ReadOnly	Tanpa pengaturan ODBC ini, tugas replikasi dirutekan ke replika grup ketersediaan utama. Untuk informasi selengkapnya, lihat Dukungan Klien Asli SQL Server untuk Ketersediaan

Atribut koneksi tambahan	Nilai	Catatan
		Tinggi, Pemulihan Bencana dalam dokumentasi SQL Server.
multiSubnetFailover	yes	Untuk informasi selengkapnya, lihat Dukungan Klien Asli SQL Server untuk Ketersediaan Tinggi, Pemulihan Bencana dalam dokumentasi SQL Server.
alwaysOnSecondarySyncBackupIsEnabled	false	Untuk informasi selengkapnya, lihat Pengaturan titik akhir saat menggunakan SQL Server sebagai sumber AWS DMS .
activateSafeguard	false	Untuk informasi lebih lanjut, lihat Batasan berikut ini.
setupMicrosoftForTables	false	Untuk informasi lebih lanjut, lihat Batasan berikut ini.

- Aktifkan opsi distribusi pada semua replika di grup ketersediaan Anda. Tambahkan semua node ke daftar distributor. Untuk informasi selengkapnya, lihat [Mengatur distribusi](#).
- Jalankan kueri berikut pada replika baca-tulis utama untuk mengaktifkan publikasi database Anda. Anda menjalankan query ini hanya sekali untuk database Anda.

```
sp_replicationdboption @dbname = N'<source DB name>', @optname = N'publish', @value = N'true';
```

Batasan

Berikut ini adalah batasan untuk bekerja dengan replika grup ketersediaan sekunder:

- AWS DMS tidak mendukung Safeguard saat menggunakan replika grup ketersediaan hanya-baca sebagai sumber. Untuk informasi selengkapnya, lihat [Pengaturan titik akhir saat menggunakan SQL Server sebagai sumber AWS DMS](#).

- AWS DMS tidak mendukung atribut koneksi `setUpMsCdcForTables` tambahan saat menggunakan replika grup ketersediaan hanya-baca sebagai sumber. Untuk informasi selengkapnya, lihat [Pengaturan titik akhir saat menggunakan SQL Server sebagai sumber AWS DMS](#).
- AWS DMS dapat menggunakan replika grup ketersediaan sekunder yang dikelola sendiri sebagai basis data sumber untuk replikasi berkelanjutan (ubah pengambilan data, atau CDC) mulai dari versi 3.4.7. Replika baca multi-AZ Cloud SQL Server tidak didukung. Jika Anda menggunakan versi sebelumnya AWS DMS, pastikan Anda menggunakan replika grup ketersediaan utama sebagai basis data sumber untuk CDC.

Failover ke node lain

Jika Anda menyetel atribut koneksi `ApplicationIntent` tambahan untuk titik akhir `AndaReadOnly`, AWS DMS tugas Anda akan tersambung ke node hanya-baca dengan prioritas perutean hanya-baca tertinggi. Kemudian gagal ke node hanya-baca lainnya di grup ketersediaan Anda ketika node hanya-baca prioritas tertinggi tidak tersedia. Jika Anda tidak mengatur `ApplicationIntent`, AWS DMS tugas Anda hanya terhubung ke node utama (baca/tulis) di grup ketersediaan Anda.

Persyaratan keamanan saat menggunakan SQL Server sebagai sumber AWS Database Migration Service

Akun AWS DMS pengguna harus memiliki setidaknya peran `db_owner` pengguna pada database SQL Server sumber yang Anda sambungkan.

Pengaturan titik akhir saat menggunakan SQL Server sebagai sumber AWS DMS

Anda dapat menggunakan pengaturan endpoint untuk mengkonfigurasi database sumber SQL Server Anda mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir sumber menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--microsoft-sql-server-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan endpoint yang dapat Anda gunakan dengan SQL Server sebagai sumber.

Nama	Penjelasan
ActivateSafeguard	<p>Atribut ini mengaktifkan atau menonaktifkan Safeguard . Untuk informasi tentang Safeguard, lihat berikut. SafeguardPolicy</p> <p>Nilai default: true</p> <p>Nilai valid: {false, true}</p> <p>Contoh: '{"ActivateSafeguard": true}'</p>
AlwaysOnSharedSynchedBackupIsEnabled	<p>Atribut ini menyesuaikan perilaku AWS DMS saat bermigrasi dari database sumber SQL Server yang di-host sebagai bagian dari klaster grup ketersediaan Selalu Aktif.</p> <p>AWS DMS telah meningkatkan dukungan untuk database sumber SQL Server yang dikonfigurasi untuk berjalan di klaster Selalu Aktif. Dalam kasus ini, AWS DMS mencoba untuk melacak jika backup transaksi terjadi dari node di klaster Always On selain node di mana instans basis data sumber menjadi host. Saat memulai tugas migrasi, AWS DMS mencoba terhubung ke setiap node di cluster, tetapi gagal jika tidak dapat terhubung ke salah satu node.</p> <p>Jika Anda AWS DMS perlu melakukan polling semua node di klaster Always On untuk cadangan transaksi, setel atribut ini ke. false</p> <p>Nilai default: true</p> <p>Nilai yang valid: true atau false</p> <p>Contoh: '{"AlwaysOnSharedSynchedBackupIsEnabled": false}'</p>
"ApplicationIntent": "readonly"	<p>Pengaturan atribut driver ODBC ini menyebabkan SQL Server merutekan tugas replikasi Anda ke node hanya-</p>

Nama	Penjelasan
	baca prioritas tertinggi. Tanpa pengaturan ini, SQL Server merutekan tugas replikasi Anda ke node baca-tulis utama.
EnableNonSysadminWrapper	<p>Gunakan pengaturan titik akhir ini saat Anda menyiapkan replikasi berkelanjutan pada server SQL mandiri tanpa pengguna sysadmin. Parameter ini didukung pada AWS DMS versi 3.4.7 dan yang lebih tinggi. Untuk informasi tentang pengaturan replikasi yang sedang berlangsung pada server SQL mandiri, lihat. Menyiapkan replikasi yang sedang berlangsung pada SQL Server mandiri: Tanpa peran sysadmin</p> <p>Nilai default: false</p> <p>Nilai valid: true, false</p> <p>Contoh: '{"EnableNonSysadminWrapper": true}'</p>
ExecuteTimeout	<p>Gunakan atribut koneksi tambahan (ECA) ini untuk mengatur batas waktu pernyataan klien untuk instance SQL Server, dalam hitungan detik. Nilai default adalah 60 detik.</p> <p>Contoh: '{"ExecuteTimeout": 100}'</p>
FatalOnSimpleModel	<p>Ketika diatur ke true, pengaturan ini menghasilkan kesalahan fatal ketika model pemulihan database SQL Server diatur ke simple.</p> <p>Nilai default: false</p> <p>Nilai yang valid: true atau false</p> <p>Contoh: '{"FatalOnSimpleModel": true}'</p>

Nama	Penjelasan
ForceLobLookup	<p>Memaksa pencarian LOB pada LOB sebaris.</p> <p>Nilai default: <code>false</code></p> <p>Nilai valid: <code>true, false</code></p> <p>Contoh: <code>'{"ForceLobLookup": false}'</code></p>
"MultiSubnetFailover": "Yes"	<p>Atribut driver ODBC ini membantu DMS untuk terhubung ke primer baru jika terjadi failover Grup Ketersediaan. Atribut ini dirancang untuk situasi ketika koneksi rusak atau alamat IP pendengar salah. Dalam situasi ini, AWS DMS upaya untuk terhubung ke semua alamat IP yang terkait dengan pendengar Grup Ketersediaan.</p>
ReadBackupOnly	<p>Penggunaan atribut ini membutuhkan hak istimewa sysadmin. Ketika atribut ini disetel ke Y, selama replikasi yang sedang berlangsung AWS DMS membaca perubahan hanya dari backup log transaksi dan tidak membaca dari file log transaksi aktif. Dengan menetapkan parameter ini ke Y Anda dapat mengontrol pertumbuhan berkas log transaksi aktif selama tugas replikasi beban penuh dan yang sedang berlangsung. Namun, parameter ini dapat menambahkan beberapa latensi sumber pada replikasi yang sedang berlangsung.</p> <p>Nilai yang benar: N atau Y. Default adalah N.</p> <p>Contoh: <code>'{"ReadBackupOnly": Y}'</code></p> <p>Catatan: Parameter ini tidak bekerja pada instans sumber SQL Server Amazon RDS karena cara RDS melakukan backup.</p>

Nama	Penjelasan
SafeguardPolicy	<p>Untuk kinerja optimal, AWS DMS cobalah untuk menangkap semua perubahan yang belum dibaca dari log transaksi aktif (TLOG). Namun, terkadang karena pemotongan, TLOG aktif mungkin tidak berisi semua perubahan yang belum dibaca. Ketika ini terjadi, AWS DMS mengakses cadangan log untuk menangkap perubahan yang hilang. Untuk meminimalkan kebutuhan untuk mengakses cadangan log, AWS DMS mencegah pemotongan menggunakan salah satu metode berikut:</p> <ol style="list-style-type: none">1. <code>RELY_ON_SQL_SERVER_REPLICATION_AGENT</code> (Mulai transaksi dalam database): Ini adalah default untuk AWS DMS. <p>Bila Anda menggunakan pengaturan ini, AWS DMS mengharuskan SQL Server Log Reader Agent berjalan, sehingga AWS DMS dapat memindahkan transaksi yang ditandai untuk replikasi dari TLOG aktif. Perhatikan bahwa jika Agen Pembaca Log tidak berjalan, TLOG aktif dapat menjadi penuh, menyebabkan database sumber beralih ke mode hanya-baca hingga Anda dapat menyelesaikan masalah. Jika Anda perlu mengaktifkan Microsoft Replication di database Anda untuk tujuan selain AWS DMS, maka Anda harus memilih pengaturan ini.</p> <p>Saat Anda menggunakan pengaturan ini, AWS DMS meminimalkan pembacaan cadangan log dengan membuat tabel yang disebut <code>awsdms_truncation_safeguard</code> dan mencegah pemotongan TLOG dengan meniru transaksi terbuka dalam database. Ini membuat database tidak memotong peristiwa dan memindahkannya ke log cadangan selama lima menit (secara default). Pastikan tabel tidak termasuk dalam rencana pemeliharaan apa pun, karena dapat</p>

Nama	Penjelasan
	<p>menyebabkan pekerjaan pemeliharaan gagal. Anda dapat menghapus tabel dengan aman jika tidak ada tugas yang dikonfigurasi dengan opsi <code>Start Transactions</code> database.</p> <p>2. <code>EXCLUSIVE_AUTOMATIC_TRUNCATION</code> (Gunakan secara eksklusif <code>sp_repldone</code> dengan satu tugas): Saat Anda menggunakan pengaturan ini, AWS DMS memiliki kontrol penuh atas proses agen replikasi yang menandai entri log sebagai <code>ready for truncation</code> penggunaan <code>sp_repldone</code>. Dengan pengaturan ini, AWS DMS tidak menggunakan transaksi dummy seperti pengaturan <code>RELY_ON_SQL_SERVER_REPLICATION_AGENT</code> (default). Anda hanya dapat menggunakan pengaturan ini ketika MS Replikasi tidak digunakan untuk tujuan lain selain AWS DMS pada database sumber. Juga, saat menggunakan pengaturan ini, hanya satu AWS DMS tugas yang dapat mengakses database. Jika Anda perlu menjalankan AWS DMS tugas paralel terhadap database yang sama, gunakan <code>RELY_ON_SQL_SERVER_REPLICATION_AGENT</code>.</p> <ul style="list-style-type: none"> • Pengaturan ini mengharuskan Agen Pembaca Log dihentikan di database. Jika Agen Pembaca Log berjalan saat tugas dimulai, AWS DMS tugas akan memaksanya untuk berhenti. Atau, Anda dapat menghentikan Agen Pembaca Log secara manual sebelum memulai tugas. • Saat menggunakan metode ini dengan MS-CDC, Anda harus menghentikan dan menonaktifkan penangkapan MS-CDC dan pekerjaan pembersihan MS-CDC. • Anda tidak dapat menggunakan pengaturan ini ketika pekerjaan Microsoft SQL Server Migrasi

Nama	Penjelasan
	<p>berjalan pada mesin Distributor jarak jauh, karena AWS DMS tidak memiliki akses ke mesin jarak jauh.</p> <ul style="list-style-type: none"> • <code>EXCLUSIVE_AUTOMATIC_TRUNCATION</code> tidak berfungsi pada instans sumber Amazon RDS for SQL Server karena pengguna Amazon RDS tidak memiliki akses untuk menjalankan prosedur yang disimpan. <code>sp_repldone</code> • Jika disetel <code>SafeguardPolicy</code> ke <code>EXCLUSIVE_AUTOMATIC_TRUNCATION</code> tanpa menggunakan peran <code>sysadmin</code>, Anda harus memberikan izin pada <code>dbo.sysjobs</code> objek <code>dbo.syscategories</code> dan kepada pengguna. <code>dmsuser</code> <p>Nilai default: <code>RELY_ON_SQL_SERVER_REPLICATION_AGENT</code></p> <p>Nilai valid: <code>{EXCLUSIVE_AUTOMATIC_TRUNCATION, RELY_ON_SQL_SERVER_REPLICATION_AGENT }</code></p> <p>Contoh: <code>'{"SafeguardPolicy": "EXCLUSIVE_AUTOMATIC_TRUNCATION"}'</code></p>

Nama	Penjelasan
SetupMsCdcForTables	<p>Atribut ini mengaktifkan MS-CDC untuk database sumber dan untuk tabel dalam pemetaan tugas yang tidak mengaktifkan replikasi MS. Menyetel nilai ini untuk true menjalankan prosedur <code>sp_cdc_enable_db</code> tersimpan pada database sumber, dan menjalankan prosedur <code>sp_cdc_enable_table</code> tersimpan pada setiap tabel dalam tugas yang tidak mengaktifkan replikasi MS di database sumber. Untuk informasi selengkapnya tentang mengaktifkan distribusi, lihat Menyiapkan replikasi yang sedang berlangsung pada SQL Server yang dikelola sendiri.</p> <p>Nilai valid: {true, false}</p> <p>Contoh: '{"SetupMsCdcForTables": true}'</p>
TlogAccessMode	<p>Menunjukkan mode yang digunakan untuk mengambil data CDC.</p> <p>Nilai default: PreferTlog</p> <p>Nilai valid: BackupOnly , PreferBackup , PreferTlog , TlogOnly</p> <p>Contoh: '{"TlogAccessMode": "PreferTlog"}'</p>
Use3rdPartyBackupDevice	<p>Ketika atribut ini diatur menjadi Y, AWS DMS memproses backup log transaksi pihak ketiga jika mereka dibuat dalam format asli.</p>

Jenis sumber data untuk SQL Server

Migrasi data yang menggunakan SQL Server sebagai sumber untuk AWS DMS mendukung sebagian besar tipe data SQL Server. Tabel berikut menunjukkan tipe data sumber SQL Server yang didukung saat menggunakan AWS DMS dan pemetaan default dari tipe AWS DMS data.

Untuk informasi tentang cara melihat jenis data yang dipetakan dalam target, lihat bagian titik akhir target yang Anda gunakan.

Untuk informasi tambahan tentang tipe AWS DMS data, lihat [Tipe data untuk AWS Database Migration Service](#).

Jenis data SQL Server	AWS DMS tipe data
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 dan lebih tinggi)	DATETIME
SMALLDATETIME	DATETIME
DATE	DATE
TIME	TIME
DATETIMEOFFSET	WSTRING

Jenis data SQL Server	AWS DMS tipe data
CHAR	STRING
VARCHAR	STRING
VARCHAR (max)	<p>CLOB</p> <p>TEXT</p> <p>Untuk menggunakan tipe data ini AWS DMS, Anda harus mengaktifkan penggunaan tipe data CLOB untuk tugas tertentu.</p> <p>Untuk tabel SQL Server, AWS DMS memperbarui kolom LOB di target bahkan untuk pernyataan UPDATE yang tidak mengubah nilai kolom LOB di SQL Server.</p> <p>Selama CDC, AWS DMS mendukung tipe data CLOB hanya dalam tabel yang menyertakan kunci utama.</p>
NCHAR	WSTRING
NVARCHAR (panjang)	WSTRING

Jenis data SQL Server	AWS DMS tipe data
NVARCHAR (maks)	<p>NCLOB</p> <p>NTEXT</p> <p>Untuk menggunakan tipe data ini AWS DMS, Anda harus mengaktifkan penggunaan SupportLobs untuk tugas tertentu. Untuk informasi selengkapnya tentang mengaktifkan dukungan Lob, lihat. Mengatur dukungan LOB untuk database sumber dalam suatu tugas AWS DMS</p> <p>Untuk tabel SQL Server, AWS DMS memperbarui kolom LOB di target bahkan untuk pernyataan UPDATE yang tidak mengubah nilai kolom LOB di SQL Server.</p> <p>Selama CDC, AWS DMS mendukung tipe data CLOB hanya dalam tabel yang menyertakan kunci utama.</p>
BINARY	BYTES
VARBINARY	BYTES

Jenis data SQL Server	AWS DMS tipe data
VARBINARY (max)	<p>BLOB</p> <p>CITRA</p> <p>Untuk tabel SQL Server, AWS DMS memperbarui kolom LOB di target bahkan untuk pernyataan UPDATE yang tidak mengubah nilai kolom LOB di SQL Server.</p> <p>Untuk menggunakan tipe data ini AWS DMS, Anda harus mengaktifkan penggunaan tipe data BLOB untuk tugas tertentu.</p> <p>AWS DMS mendukung tipe data BLOB hanya dalam tabel yang menyertakan kunci utama.</p>
TIMESTAMP	BYTES
UNIQUEIDENTIFIER	STRING
HIERARCHYID	<p>Gunakan HIERARCHYID saat mereplikasi ke titik akhir target SQL Server.</p> <p>Gunakan WSTRING (250) ketika mereplikasi ke semua titik akhir target lainnya.</p>

Jenis data SQL Server	AWS DMS tipe data
XML	<p data-bbox="833 226 943 258">NCLOB</p> <p data-bbox="833 306 1430 485">Untuk tabel SQL Server, AWS DMS memperbarui kolom LOB di target bahkan untuk pernyataan UPDATE yang tidak mengubah nilai kolom LOB di SQL Server.</p> <p data-bbox="833 531 1471 659">Untuk menggunakan tipe data ini AWS DMS, Anda harus mengaktifkan penggunaan tipe data NCLOB untuk tugas tertentu.</p> <p data-bbox="833 705 1495 833">Selama CDC, AWS DMS mendukung tipe data NCLOB hanya dalam tabel yang menyertakan kunci utama.</p>
GEOMETRY	<p data-bbox="833 884 1495 961">Gunakan GEOMETRY ketika mereplikasi ke titik akhir target yang mendukung jenis data ini.</p> <p data-bbox="833 1008 1495 1085">Gunakan CLOB ketika mereplikasi ke titik akhir target yang tidak mendukung jenis data ini.</p>
GEOGRAPHY	<p data-bbox="833 1142 1495 1220">Gunakan GEOGRAPHY ketika mereplikasi ke titik akhir target yang mendukung jenis data ini.</p> <p data-bbox="833 1266 1495 1344">Gunakan CLOB ketika mereplikasi ke titik akhir target yang tidak mendukung jenis data ini.</p>

AWS DMS tidak mendukung tabel yang menyertakan bidang dengan tipe data berikut.

- CURSOR
- SQL_VARIANT
- TABLE

Note

Jenis data yang ditetapkan pengguna didukung sesuai dengan jenis dasar mereka. Misalnya, jenis data yang ditetapkan pengguna berdasarkan DATETIME diperlakukan sebagai jenis data DATETIME.

Menggunakan basis data Microsoft Azure SQL sebagai sumber untuk AWS DMS

Dengan AWS DMS, Anda dapat menggunakan Microsoft Azure SQL Database sebagai sumber sama seperti SQL Server. AWS DMS mendukung, sebagai sumber, daftar versi basis data yang sama yang didukung untuk SQL Server yang berjalan lokal atau instans Amazon EC2.

Untuk informasi selengkapnya, lihat [Menggunakan database Microsoft SQL Server sebagai sumber AWS DMS](#).

Note

AWS DMS tidak mendukung operasi change data capture (CDC) dengan Azure SQL Database.

Menggunakan Microsoft Azure SQL Managed Instance sebagai sumber untuk AWS DMS

Dengan AWS DMS, Anda dapat menggunakan Microsoft Azure SQL Managed Instance sebagai sumber dengan cara yang hampir sama seperti SQL Server. AWS DMS mendukung, sebagai sumber, daftar versi database yang sama yang didukung untuk SQL Server yang berjalan di tempat atau pada instans Amazon EC2.

Untuk informasi selengkapnya, lihat [Menggunakan database Microsoft SQL Server sebagai sumber AWS DMS](#).

Menggunakan Microsoft Azure Database untuk PostgreSQL server fleksibel sebagai sumber untuk AWS DMS

Dengan AWS DMS, Anda dapat menggunakan Microsoft Azure Database untuk PostgreSQL server fleksibel sebagai sumber dalam banyak cara yang sama seperti yang Anda lakukan PostgreSQL.

Untuk informasi tentang versi Microsoft Azure Database untuk PostgreSQL server fleksibel AWS DMS yang mendukung sebagai sumber, lihat [Sumber untuk AWS DMS](#)

Menyiapkan Microsoft Azure untuk PostgreSQL server fleksibel untuk replikasi logis dan decoding

Anda dapat menggunakan replikasi logis dan fitur decoding di Microsoft Azure Database untuk PostgreSQL server fleksibel selama migrasi database.

Untuk decoding logis, DMS menggunakan plugin atau `test_decoding` `pglogical`. Jika `pglogical` plugin tersedia pada database PostgreSQL sumber, DMS membuat slot `pglogical` replikasi menggunakan, jika tidak plugin digunakan. `test_decoding`

Untuk mengonfigurasi server fleksibel Microsoft Azure untuk PostgreSQL Anda sebagai titik akhir sumber untuk DMS, lakukan langkah-langkah berikut:

1. Buka halaman Parameter Server di portal.
2. Atur parameter `wal_level` server ke `LOGICAL`.
3. Jika Anda ingin menggunakan `pglogical` ekstensi, atur `azure.extensions` parameter `shared_preload_libraries` dan `kepglogical`.
4. Atur `max_replication_slots` parameter ke jumlah maksimum tugas DMS yang Anda rencanakan untuk dijalankan secara bersamaan. Di Microsoft Azure, nilai default untuk parameter ini adalah 10. Nilai maksimum parameter ini bergantung pada memori yang tersedia dari instance PostgreSQL Anda, memungkinkan antara 2 dan 8 slot replikasi per GB memori.
5. Atur `max_wal_senders` parameter ke nilai yang lebih besar dari 1. Parameter `max_wal_senders` mengatur jumlah tugas bersamaan yang dapat berjalan. Nilai default adalah 10.
6. Tetapkan nilai `max_worker_processes` parameter setidaknya 16. Jika tidak, Anda mungkin melihat kesalahan seperti berikut:

```
WARNING: out of background worker slots.
```

7. Simpan perubahan. Mulai ulang server untuk menerapkan perubahan.
8. Konfirmasikan bahwa instance PostgreSQL Anda memungkinkan lalu lintas jaringan dari sumber daya penghubung Anda.
9. Berikan izin replikasi pengguna yang ada, atau buat pengguna baru dengan izin replikasi, menggunakan perintah berikut.
 - Berikan izin replikasi pengguna yang ada menggunakan perintah berikut:

```
ALTER USER <existing_user> WITH REPLICATION;
```

- Buat pengguna baru dengan izin replikasi menggunakan perintah berikut:

```
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;
```

Untuk informasi selengkapnya tentang replikasi logis dengan PostgreSQL, lihat topik berikut:

- [Mengaktifkan change data capture \(CDC\) menggunakan replikasi logis](#)
- [Menggunakan titik awal CDC asli untuk mengatur beban CDC dari sumber PostgreSQL](#)
- [Replikasi logis dan decoding logis di Azure Database untuk PostgreSQL - Server Fleksibel dalam Database Azure untuk dokumentasi PostgreSQL.](#)

Menggunakan Microsoft Azure Database untuk MySQL server fleksibel sebagai sumber AWS DMS

Dengan AWS DMS, Anda dapat menggunakan Microsoft Azure Database untuk MySQL server fleksibel sebagai sumber dalam banyak cara yang sama seperti yang Anda lakukan MySQL.

Untuk informasi tentang versi Microsoft Azure Database untuk server fleksibel MySQL AWS DMS yang mendukung sebagai sumber, lihat. [Sumber untuk AWS DMS](#)

Untuk informasi selengkapnya tentang menggunakan database yang kompatibel dengan MySQL yang dikelola pelanggan, lihat. AWS DMS [Menggunakan database yang kompatibel dengan MySQL yang dikelola sendiri sebagai sumber AWS DMS](#)

Keterbatasan saat menggunakan Azure MySQL sebagai sumber AWS Database Migration Service

- Nilai default untuk variabel sistem server fleksibel Azure MySQL `sql_generate_invisible_primary_key` adalah `0N`, dan server secara otomatis menambahkan kunci primer tak terlihat (GIPK) yang dihasilkan ke tabel apa pun yang dibuat tanpa kunci primer eksplisit. AWS DMS tidak mendukung replikasi berkelanjutan untuk tabel MySQL dengan batasan GIPK.

Menggunakan OCI MySQL Heatwave sebagai sumber untuk AWS DMS

Dengan AWS DMS, Anda dapat menggunakan OCI MySQL Heatwave sebagai sumber dalam banyak cara yang sama seperti yang Anda lakukan MySQL. Menggunakan OCI MySQL Heatwave sebagai sumber memerlukan beberapa perubahan konfigurasi tambahan.

Untuk informasi tentang versi OCI MySQL Heatwave AWS DMS yang mendukung sebagai sumber, lihat [Sumber untuk AWS DMS](#)

Menyiapkan OCI MySQL Heatwave untuk replikasi logis

Untuk mengonfigurasi instance OCI MySQL Heatwave Anda sebagai titik akhir sumber untuk DMS, lakukan hal berikut:

1. Masuk ke OCI Console, dan buka menu hamburger utama (≡) di sudut kiri atas.
2. Pilih Database, Sistem DB.
3. Buka menu Konfigurasi.
4. Pilih Buat konfigurasi.
5. Masukkan nama konfigurasi, seperti **`dms_configuration`**.
6. Pilih bentuk instance OCI MySQL Heatwave Anda saat ini. Anda dapat menemukan bentuk pada tab properti konfigurasi sistem DB instans di bawah bagian DB system Configuration:Shape.
7. Di bagian Variabel pengguna, pilih variabel `binlog_row_value_options` sistem. Nilai defaultnya adalah `PARTIAL_JSON`. Hapus nilainya.
8. Pilih tombol Buat.
9. Buka instance OCI MySQL HeatWave Anda, dan pilih tombol Edit.
10. Di bagian Konfigurasi, pilih tombol Ubah konfigurasi, dan pilih konfigurasi bentuk yang Anda buat di langkah 4.

11. Setelah perubahan diterapkan, instance Anda siap untuk replikasi logis.

Menggunakan Google Cloud untuk MySQL sebagai sumber untuk AWS DMS

Dengan AWS DMS, Anda dapat menggunakan Google Cloud untuk MySQL sebagai sumber dengan cara yang sama seperti yang Anda lakukan MySQL.

Untuk informasi tentang versi GCP MySQL AWS DMS yang mendukung sebagai sumber, lihat [Sumber untuk AWS DMS](#)

Untuk informasi selengkapnya, lihat [Menggunakan basis data yang kompatibel dengan MySQL sebagai sumber untuk AWS DMS](#).

Note

Support untuk GCP MySQL 8.0 sebagai sumber tersedia dalam versi 3.4.6. AWS DMS tidak mendukung mode `SSL verify-full` untuk GCP untuk instance MySQL. `Allow only SSL connections` Pengaturan keamanan MySQL GCP tidak didukung, karena memerlukan verifikasi sertifikat server dan klien. AWS DMS hanya mendukung verifikasi sertifikat server. AWS DMS mendukung default GCP CloudSQL untuk nilai MySQL untuk bendera database. `CRC32 binlog_checksum`

Menggunakan Google Cloud untuk PostgreSQL sebagai sumber untuk AWS DMS

Dengan AWS DMS, Anda dapat menggunakan Google Cloud untuk PostgreSQL sebagai sumber dengan cara yang sama seperti database PostgreSQL yang dikelola sendiri.

Untuk informasi tentang versi GCP PostgreSQL AWS DMS yang mendukung sebagai sumber, lihat [Sumber untuk AWS DMS](#)

Untuk informasi selengkapnya, lihat [Menggunakan basis data PostgreSQL sebagai sumber AWS DMS](#).

Siapkan Google Cloud untuk PostgreSQL untuk replikasi logis dan decoding

Anda dapat menggunakan fitur replikasi dan decoding logis di Google Cloud SQL untuk PostgreSQL selama migrasi database.

Untuk decoding logis, DMS menggunakan salah satu plugin berikut:

- `test_decoding`
- `pglogical`

Jika `pglogical` plugin tersedia pada database PostgreSQL sumber, DMS membuat slot `pglogical` replikasi menggunakan, jika tidak plugin digunakan. `test_decoding`

Perhatikan hal berikut tentang menggunakan decoding logis dengan: AWS DMS

1. Dengan Google Cloud SQL untuk PostgreSQL, aktifkan decoding logis dengan menyetel flag ke `cloudsql.logical_decoding on`
2. Untuk mengaktifkan `pglogical`, atur `cloudsql.enable_pglogical` flag ke on, dan restart database.
3. Untuk menggunakan fitur decoding logis, Anda membuat pengguna PostgreSQL dengan atribut `REPLICATION`. Saat Anda menggunakan `pglogical` ekstensi, pengguna harus memiliki `cloudsqlsuperuser` peran. Untuk membuat pengguna dengan `cloudsqlsuperuser` peran, lakukan hal berikut:

```
CREATE USER new_aws_dms_user WITH REPLICATION
IN ROLE cloudsqlsuperuser LOGIN PASSWORD 'new_aws_dms_user_password';
```

Untuk menyetel atribusi ini pada pengguna yang sudah ada, lakukan hal berikut:

```
ALTER USER existing_user WITH REPLICATION;
```

4. Atur `max_replication_slots` parameter ke jumlah maksimum tugas DMS yang Anda rencanakan untuk dijalankan secara bersamaan. Di Google Cloud SQL, nilai default untuk parameter ini adalah 10. Nilai maksimum parameter ini bergantung pada memori yang tersedia dari instance PostgreSQL Anda, memungkinkan antara 2 dan 8 slot replikasi per GB memori.

Untuk informasi selengkapnya tentang replikasi logis dengan PostgreSQL, lihat topik berikut:

- [Mengaktifkan change data capture \(CDC\) menggunakan replikasi logis](#)
- [Menggunakan titik awal CDC asli untuk mengatur beban CDC dari sumber PostgreSQL](#)
- [Siapkan replikasi logis dan decoding di Cloud SQL untuk dokumentasi PostgreSQL.](#)

Menggunakan basis data PostgreSQL sebagai sumber AWS DMS

Anda dapat memigrasi data dari satu atau banyak basis data PostgreSQL menggunakan AWS DMS. Dengan database PostgreSQL sebagai sumber, Anda dapat memigrasikan data ke database PostgreSQL lain atau salah satu database lain yang didukung.

Untuk informasi tentang versi PostgreSQL AWS DMS yang mendukung sebagai sumber, lihat. [Sumber untuk AWS DMS](#)

AWS DMS mendukung PostgreSQL untuk jenis database ini:

- Basis data lokal
- Basis data pada instans Amazon EC2
- Basis data pada instans DB Amazon RDS
- Basis data pada instans DB berdasarkan Amazon Aurora Edisi Kompatibel PostgreSQL
- Database pada instans DB berdasarkan Amazon Aurora PostgreSQL yang kompatibel dengan Serverless Edition

Note

DMS mendukung Amazon Aurora PostgreSQL — V1 tanpa server sebagai sumber untuk Full load saja. Tetapi Anda dapat menggunakan Amazon Aurora PostgreSQL—Serverless V2 sebagai sumber untuk tugas Full load, Full load+CDC, dan CDC saja.

Versi AWS DMS yang digunakan

Gunakan AWS DMS versi apapun yang tersedia.

Versi AWS DMS yang digunakan

Gunakan AWS DMS versi 3.4.3 dan lebih tinggi.

Gunakan AWS DMS versi 3.4.7 dan lebih tinggi.

Gunakan AWS DMS versi 3.5.1 dan lebih tinggi.

Anda dapat menggunakan Lapisan Soket Aman (SSL) untuk mengenkripsi sambungan antara titik akhir PostgreSQL Anda dan instans replikasi. Untuk informasi selengkapnya tentang penggunaan SSL dengan titik akhir PostgreSQL, lihat [Menggunakan SSL dengan AWS Database Migration Service](#).

Sebagai persyaratan keamanan tambahan saat menggunakan PostgreSQL sebagai sumber, akun pengguna yang ditentukan harus menjadi pengguna terdaftar di basis data PostgreSQL.

Untuk mengonfigurasi basis data PostgreSQL sebagai titik akhir sumber AWS DMS, lakukan hal berikut:

- Buat pengguna PostgreSQL dengan izin yang sesuai untuk memberikan AWS DMS akses ke basis data sumber PostgreSQL Anda.

Note

- Jika basis data sumber PostgreSQL Anda dikelola sendiri, lihat [Menggunakan basis data PostgreSQL yang dikelola sendiri sebagai sumber di AWS DMS](#) untuk informasi lebih lanjut.
 - Jika basis data sumber PostgreSQL Anda dikelola oleh Amazon RDS, lihat [Bekerja dengan database PostgreSQL AWS -managed sebagai sumber DMS](#) untuk informasi lebih lanjut.
- Buat titik akhir sumber PostgreSQL yang sesuai dengan konfigurasi basis data PostgreSQL yang Anda pilih.

- Buat tugas atau serangkaian tugas untuk memigrasi tabel Anda.

Untuk membuat full-load-only tugas, tidak diperlukan konfigurasi titik akhir lebih lanjut.

Sebelum Anda membuat tugas untuk change data capture (tugas CDC saja atau tugas beban penuh dan CDC), lihat [Mengaktifkan CDC menggunakan basis data PostgreSQL yang dikelola sendiri sebagai sumber AWS DMS](#) atau [Mengaktifkan CDC dengan instans DB PostgreSQL terkelola AWS dengan AWS DMS](#).

Topik

- [Menggunakan basis data PostgreSQL yang dikelola sendiri sebagai sumber di AWS DMS](#)
- [Bekerja dengan database PostgreSQL AWS -managed sebagai sumber DMS](#)
- [Mengaktifkan change data capture \(CDC\) menggunakan replikasi logis](#)
- [Menggunakan titik awal CDC asli untuk mengatur beban CDC dari sumber PostgreSQL](#)
- [Bermigrasi dari PostgreSQL ke PostgreSQL menggunakan AWS DMS](#)
- [Menghapus artefak AWS DMS dari basis data sumber PostgreSQL](#)
- [Pengaturan konfigurasi tambahan ketika menggunakan basis data PostgreSQL sebagai sumber DMS](#)
- [Menggunakan pengaturan titik akhir MapBooleanAsBoolean PostgreSQL](#)
- [Pengaturan titik akhir dan Atribut Koneksi Ekstra \(ECA\) saat menggunakan PostgreSQL sebagai sumber DMS](#)
- [Keterbatasan menggunakan basis data PostgreSQL sebagai sumber DMS](#)
- [Jenis data sumber untuk PostgreSQL](#)

Menggunakan basis data PostgreSQL yang dikelola sendiri sebagai sumber di AWS DMS

Dengan basis data PostgreSQL yang dikelola sendiri sebagai sumber, Anda dapat memigrasi data ke basis data PostgreSQL lain, atau salah satu basis data target lain yang didukung oleh AWS DMS. Sumber basis data dapat berupa basis data lokal atau mesin yang dikelola sendiri yang berjalan pada instans Amazon EC2. Anda dapat menggunakan instans DB untuk baik tugas beban penuh maupun tugas chane data capture (CDC).

Prasyarat untuk menggunakan basis data PostgreSQL yang dikelola sendiri sebagai sumber AWS DMS

Sebelum memigrasi data dari basis data sumber PostgreSQL yang dikelola sendiri, lakukan hal berikut:

- Pastikan Anda menggunakan database PostgreSQL yang versi 9.4.x atau lebih tinggi.
- Untuk tugas beban penuh plus tugas CDC atau tugas CDC saja, berikan izin pengguna super untuk akun pengguna yang ditentukan untuk basis data sumber PostgreSQL. Akun pengguna membutuhkan izin pengguna super untuk mengakses fungsi khusus replikasi dalam sumber. Untuk tugas beban penuh beban saja, akun pengguna perlu izin SELECT pada tabel untuk memigrasikannya.
- Tambahkan alamat IP dari server replikasi AWS DMS untuk file konfigurasi `pg_hba.conf` dan mengaktifkan replikasi dan sambungan socket. Lihat contoh berikut.

```
# 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
```

File konfigurasi `pg_hba.conf` PostgreSQL mengontrol autentikasi klien. (HBA adalah singkatan dari host-based authentication.) File secara tradisional disimpan dalam direktori data kluster basis data.

- Jika Anda mengonfigurasi basis data sebagai sumber untuk replikasi logis menggunakan AWS DMS lihat [Mengaktifkan CDC menggunakan basis data PostgreSQL yang dikelola sendiri sebagai sumber AWS DMS](#).

Note

Beberapa transaksi AWS DMS tidak jalan untuk beberapa waktu sebelum mesin DMS menggunakannya lagi. Dengan menggunakan parameter `idle_in_transaction_session_timeout` di PostgreSQL versi 9.6 dan yang lebih tinggi, Anda dapat menyebabkan transaksi idle habis dan gagal. Jangan mengakhiri transaksi yang diam saat Anda menggunakan AWS DMS.

Mengaktifkan CDC menggunakan basis data PostgreSQL yang dikelola sendiri sebagai sumber AWS DMS.

AWS DMS mendukung change data capture (CDC) menggunakan replikasi logis. Untuk mengaktifkan replikasi logis dari basis data sumber PostgreSQL yang dikelola sendiri, tetapkan parameter dan nilai berikut di file `postgresql.conf` konfigurasi:

- Tetapkan `wal_level = logical`.
- Tetapkan `max_replication_slots` untuk nilai yang lebih besar dari 1.

Tetapkan nilai `max_replication_slots` sesuai dengan jumlah tugas yang ingin Anda jalankan. Misalnya, untuk menjalankan lima tugas Anda menetapkan minimal lima slot. Slot terbuka secara otomatis segera setelah tugas dimulai dan tetap terbuka bahkan ketika tugas tidak lagi berjalan. Pastikan Anda menghapus slot terbuka secara manual. Perhatikan bahwa DMS secara otomatis menjatuhkan slot replikasi ketika tugas dihapus, jika DMS membuat slot.

- Atur `max_wal_senders` menjadi nilai yang lebih besar dari 1.

Parameter `max_wal_senders` mengatur jumlah tugas bersamaan yang dapat berjalan.

- Parameter `wal_sender_timeout` mengakhiri sambungan replikasi yang tidak aktif lebih lama dari jumlah milidetik tertentu. Default untuk database PostgreSQL lokal adalah 60000 milidetik (60 detik). Menyetel nilai ke 0 (nol) menonaktifkan mekanisme batas waktu, dan merupakan pengaturan yang valid untuk DMS.

Saat menyetel `wal_sender_timeout` ke nilai bukan nol, tugas DMS dengan CDC membutuhkan minimal 10.000 milidetik (10 detik), dan gagal jika nilainya kurang dari 10.000. Jaga nilai kurang dari 5 menit untuk menghindari penundaan selama failover multi-AZ dari instance replikasi DMS.

Beberapa parameter bersifat statis, dan Anda hanya dapat mengaturnya pada awal server. Setiap perubahan pada entri mereka dalam file konfigurasi (untuk database yang dikelola sendiri) atau grup parameter DB (untuk RDS untuk database PostgreSQL) diabaikan sampai server dimulai ulang. Untuk informasi selengkapnya, lihat [dokumentasi PostgreSQL](#).

Untuk informasi lebih lanjut tentang pengaktifan CDC, lihat [Mengaktifkan change data capture \(CDC\) menggunakan replikasi logis](#).

Bekerja dengan database PostgreSQL AWS -managed sebagai sumber DMS

Anda dapat menggunakan instans DB PostgreSQL terkelola AWS sebagai sumber untuk AWS DMS. Anda dapat melakukan tugas beban penuh dan tugas change data capture (CDC) dengan menggunakan sumber PostgreSQL terkelola AWS.

Prasyarat untuk menggunakan basis data PostgreSQL terkelola AWS sebagai sumber DMS

Sebelum memigrasi data dari basis data sumber PostgreSQL terkelola AWS, lakukan hal berikut:

- Sebaiknya gunakan akun AWS pengguna dengan izin minimum yang diperlukan untuk instans PostgreSQL DB sebagai akun pengguna untuk titik akhir sumber PostgreSQL. AWS DMS Menggunakan akun master tidak disarankan. Akun harus memiliki `rds_superuser` peran dan `rds_replication` peran. Peran `rds_replication` memberikan izin untuk mengelola slot logis dan mengalirkan data menggunakan slot logis.

Pastikan untuk membuat beberapa objek dari akun pengguna master untuk akun yang Anda gunakan. Untuk informasi tentang pembuatan objek ini, lihat [Memigrasikan basis data Amazon RDS for PostgreSQL tanpa menggunakan akun pengguna utama](#).

- Jika basis data sumber Anda dalam virtual private cloud (VPC), pilih grup keamanan VPC yang menyediakan akses ke instans DB di mana basis data berada. Hal ini diperlukan untuk instans replikasi DMS untuk menghubungkan berhasil ke instans DB sumber. Ketika instans replikasi basis data dan DMS dalam VPC yang sama, tambahkan grup keamanan yang sesuai untuk aturan inbound-nya sendiri.

Note

Beberapa transaksi AWS DMS tidak jalan untuk beberapa waktu sebelum mesin DMS menggunakannya lagi. Dengan menggunakan parameter `idle_in_transaction_session_timeout` di PostgreSQL versi 9.6 dan yang lebih tinggi, Anda dapat menyebabkan transaksi idle habis dan gagal. Jangan mengakhiri transaksi yang saat Anda menggunakan AWS DMS.

Mengaktifkan CDC dengan instans DB PostgreSQL terkelola AWS dengan AWS DMS

AWS DMS mendukung CDC pada basis data Amazon RDS PostgreSQL ketika instans DB dikonfigurasi untuk menggunakan replikasi logis. Tabel berikut merangkum kompatibilitas replikasi logis dari setiap versi PostgreSQL terkelola AWS.

Anda tidak dapat menggunakan RDS PostgreSQL untuk membaca replika untuk CDC (replikasi yang sedang berlangsung).

Versi PostgreSQL	Dukungan beban penuh AWS DMS	Dukungan CDC AWS DMS
Aurora PostgreSQL versi 2.1 ini kompatibel dengan PostgreSQL 10.5 (atau versi di bawahnya).	Ya	Tidak
Aurora PostgreSQL versi 2.2 dengan kompatibilitas PostgreSQL 10.6 (atau lebih tinggi)	Ya	Ya
RDS untuk PostgreSQL dengan kompatibilitas PostgreSQL 10.21 (atau lebih tinggi)	Ya	Ya

Mengaktifkan replikasi logis untuk instans DB RDS PostgreSQL

- Gunakan akun pengguna utama AWS untuk instans DB PostgreSQL sebagai akun pengguna untuk titik akhir sumber PostgreSQL. Akun pengguna utama memiliki peran yang diperlukan yang memungkinkan untuk mengatur CDC.

Jika Anda menggunakan akun selain akun pengguna utama, pastikan untuk membuat beberapa objek dari akun master untuk akun yang Anda gunakan. Untuk informasi selengkapnya, lihat [Memigrasikan basis data Amazon RDS for PostgreSQL tanpa menggunakan akun pengguna utama](#).

2. Atur parameter `rds.logical_replication` dalam grup parameter DB KLASER Anda menjadi 1. Parameter statis memerlukan reboot dari instans DB agar menjadi berpengaruh. Sebagai bagian dari penerapan parameter ini, AWS DMS menetapkan parameter `wal_level`, `max_wal_senders`, `max_replication_slots`, dan `max_connections`. Perubahan parameter ini dapat meningkatkan generasi write ahead log (WAL), jadi atur `rds.logical_replication` saja ketika Anda menggunakan slot replikasi logis.
3. Parameter `wal_sender_timeout` mengakhiri sambungan replikasi yang tidak aktif lebih lama dari jumlah milidetik tertentu. Default untuk database PostgreSQL AWS -managed adalah 30000 milidetik (30 detik). Menyetel nilai ke 0 (nol) menonaktifkan mekanisme batas waktu, dan merupakan pengaturan yang valid untuk DMS.

Saat menyetel `wal_sender_timeout` ke nilai bukan nol, tugas DMS dengan CDC membutuhkan minimal 10.000 milidetik (10 detik), dan gagal jika nilainya antara 0 dan 10000. Jaga nilai kurang dari 5 menit untuk menghindari penundaan selama failover multi-AZ dari instance replikasi DMS.

4. Pastikan nilai `max_worker_processes` parameter dalam Grup Parameter Cluster DB Anda sama dengan, atau lebih tinggi dari total nilai gabungan `max_logical_replication_workers`, `autovacuum_max_workers`, dan `max_parallel_workers`. Sejumlah besar proses pekerja latar belakang dapat memengaruhi beban kerja aplikasi pada instance kecil. Jadi, pantau kinerja database Anda jika Anda menetapkan `max_worker_processes` lebih tinggi dari nilai default.

Memigrasikan basis data Amazon RDS for PostgreSQL tanpa menggunakan akun pengguna utama

Dalam beberapa kasus, Anda mungkin tidak menggunakan akun pengguna utama untuk instans DB Amazon RDS PostgreSQL yang Anda gunakan sebagai sumber. Dalam kasus ini, Anda membuat beberapa objek untuk menangkap peristiwa data definition language (DDL). Anda membuat objek ini di akun selain akun utama kemudian membuat pemicu di akun pengguna utama.

Note

Jika Anda mengatur pengaturan `captureDDLs` titik akhir ke `false` titik akhir sumber, Anda tidak perlu membuat tabel berikut dan memicu pada database sumber.

Gunakan prosedur berikut untuk membuat objek-objek ini.

Membuat objek

1. Pilih skema di tempat objek akan dibuat. Skema default adalah `public`. Pastikan bahwa skema ada dan dapat diakses oleh akun *OtherThanMaster*.
2. Masuk ke instans DB PostgreSQL menggunakan akun pengguna selain akun master, berikut akun *OtherThanMaster*.
3. Buat tabel `awsdms_ddl_audit` dengan menjalankan perintah berikut, mengganti *objects_schema* dalam kode berikut dengan nama skema untuk yang digunakan.

```
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. Buat fungsi `awsdms_intercept_ddl` dengan menjalankan perintah berikut, mengganti *objects_schema* dalam kode berikut dengan nama skema untuk yang digunakan.

```
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. Keluar dari akun *OtherThanMaster* dan masuk dengan akun yang memiliki peran `rds_superuser` yang ditentukan akun itu.
6. Buat pemicu peristiwa `awsdms_intercept_ddl` dengan menjalankan perintah berikut.

```

CREATE EVENT TRIGGER awsdms_intercept_ddl ON ddl_command_end
EXECUTE PROCEDURE objects_schema.awsdms_intercept_ddl();

```

7. Pastikan bahwa semua pengguna dan peran yang mengakses peristiwa ini memiliki izin DDL yang diperlukan. Sebagai contoh:

```

grant all on public.awsdms_ddl_audit to public;
grant all on public.awsdms_ddl_audit_c_key_seq to public;

```

Ketika Anda telah menyelesaikan prosedur sebelumnya, Anda dapat membuat titik akhir sumber AWS DMS menggunakan akun *OtherThanMaster*.

Note

Peristiwa ini dipicu oleh pernyataan `CREATE TABLE`, `ALTER TABLE`, dan `DROP TABLE`.

Mengaktifkan change data capture (CDC) menggunakan replikasi logis

Anda dapat menggunakan fitur replikasi logis asli PostgreSQL untuk mengaktifkan change data capture (CDC) selama migrasi basis data untuk sumber PostgreSQL. Anda dapat menggunakan fitur ini dengan PostgreSQL yang dikelola sendiri dan juga instans Amazon RDS for PostgreSQL SQL DB.

Pendekatan ini mengurangi downtime dan membantu memastikan bahwa basis data target sinkron dengan basis data PostgreSQL sumber.

AWS DMS mendukung CDC untuk tabel PostgreSQL dengan kunci primer. Jika tabel tidak memiliki kunci primer, write ahead log (WAL) tidak mencakup gambar sebelum baris basis data. Dalam kasus ini, DMS tidak dapat memperbarui tabel. Di sini, Anda dapat menggunakan pengaturan konfigurasi tambahan dan menggunakan identitas replika tabel sebagai solusi. Namun, pendekatan ini dapat menghasilkan log tambahan. Kami merekomendasikan agar Anda menggunakan identitas replika tabel sebagai solusi hanya setelah pengujian yang cermat. Untuk informasi selengkapnya, lihat [Pengaturan konfigurasi tambahan ketika menggunakan basis data PostgreSQL sebagai sumber DMS](#).

Note

REPLICA IDENTITY FULL didukung dengan plugin decoding logis, tetapi tidak didukung dengan plugin pglogical. Untuk informasi lebih lanjut, lihat dokumentasi [pglogical](#).

Untuk tugas beban penuh dan CDC dan tugas CDC saja, AWS DMS menggunakan slot replikasi logis untuk mempertahankan log WAL untuk replikasi sampai log diterjemahkan. Pada saat mengulang kembali (bukan saat melanjutkan) untuk tugas beban penuh dan CDC atau tugas CDC, slot replikasi akan diciptakan kembali.

Note

Untuk decoding logis, DMS menggunakan test_decoding atau pglogical plugin. Jika plugin pglogical tersedia pada basis data PostgreSQL sumber, DMS menciptakan slot replikasi menggunakan pglogical, jika tidak plugin test_decoding digunakan. [Untuk informasi selengkapnya tentang plugin test_decoding, lihat Dokumentasi PostgreSQL.](#)

Jika parameter database max_slot_wal_keep_size diatur ke nilai non default, dan slot replikasi berada di belakang LSN saat ini lebih dari ukuran ini, tugas DMS gagal karena penghapusan file WAL yang diperlukan. `restart_lsn`

Mengonfigurasi plugin pglogical

Diimplementasikan sebagai ekstensi PostgreSQL, plugin pglogical adalah sistem replikasi logis dan model untuk replikasi data selektif. Tabel berikut mengidentifikasi versi basis data PostgreSQL sumber yang mendukung plugin pglogical.

Sumber PostgreSQL	Mendukung pglogical
PostgreSQL 9.4 atau versi yang lebih tinggi yang dikelola sendiri	Ya
Amazon RDS PostgreSQL 9.5 atau versi yang lebih rendah	Tidak
Amazon RDS PostgreSQL 9.6 atau versi yang lebih tinggi	Ya
Aurora PostgreSQL 1.x sampai 2.5.x	Tidak
Aurora PostgreSQL 2.6.x atau versi yang lebih tinggi	Ya
Aurora PostgreSQL 3.3.x atau versi yang lebih tinggi	Ya

Sebelum mengonfigurasi pglogical untuk digunakan dengan AWS DMS, pertama-tama aktifkan replikasi logis untuk change data capture (CDC) pada basis data sumber PostgreSQL Anda.

- Untuk informasi tentang pengaktifan replikasi logis untuk CDC pada basis data sumber PostgreSQL yang dikelola sendiri, lihat [Mengaktifkan CDC menggunakan basis data PostgreSQL yang dikelola sendiri sebagai sumber AWS DMS](#).
- Untuk informasi tentang pengaktifan replikasi logis untuk CDC pada basis data sumber PostgreSQL terkelola AWS, lihat [Mengaktifkan CDC dengan instans DB PostgreSQL terkelola AWS dengan AWS DMS](#).

Setelah replikasi logis diaktifkan pada basis data sumber PostgreSQL Anda, gunakan langkah-langkah berikut untuk mengonfigurasi pglogical untuk digunakan dengan DMS.

Menggunakan plugin pglogical untuk replikasi logis pada basis data sumber PostgreSQL dengan AWS DMS

1. Buat ekstensi pglogical pada basis data PostgreSQL sumber Anda:
 - a. Atur parameter yang benar:

- Untuk basis data PostgreSQL yang dikelola sendiri, atur parameter basis data `shared_preload_libraries= 'pglogical'`.
 - Untuk basis data PostgreSQL pada Amazon RDS dan Amazon Aurora Edisi Kompatibel PostgreSQL, atur parameter `shared_preload_libraries` menjadi `pglogical` dalam grup parameter RDS yang sama.
- b. Mulai ulang basis data sumber PostgreSQL Anda.
 - c. Pada basis data PostgreSQL, jalankan perintah, `create extension pglogical;`
2. Jalankan perintah berikut untuk melakukan verifikasi bahwa `pglogical` berhasil diinstal:

```
select * FROM pg_catalog.pg_extension
```

Sekarang Anda dapat membuat tugas AWS DMS yang melakukan change data capture untuk titik akhir basis data sumber PostgreSQL Anda.

Note

Jika Anda tidak mengaktifkan `pglogical` pada basis data sumber PostgreSQL Anda, AWS DMS menggunakan plugin `test_decoding` secara default. Ketika `pglogical` diaktifkan untuk decoding logis, AWS DMS menggunakan `pglogical` secara default. Tapi Anda dapat mengatur atribut sambungan tambahan, `PluginName` untuk menggunakan plugin `test_decoding` sebagai gantinya.

Menggunakan titik awal CDC asli untuk mengatur beban CDC dari sumber PostgreSQL

Untuk mengaktifkan titik awal CDC asli dengan PostgreSQL sebagai sumber, atur atribut sambungan tambahan `s1otName` menjadi nama slot replikasi logis yang ada ketika Anda membuat titik akhir. Slot replikasi logis ini menyimpan perubahan yang sedang berlangsung dari waktu penciptaan titik akhir, sehingga mendukung replikasi dari periode waktu sebelumnya.

PostgreSQL menulis perubahan basis data di file WAL yang dibuang hanya setelah AWS DMS berhasil membaca perubahan dari slot replikasi logis. Menggunakan slot replikasi logis dapat melindungi perubahan yang dicatat dari penghapusan sebelum slot dikonsumsi oleh mesin replikasi.

Namun, tergantung pada tingkat perubahan dan konsumsi, perubahan yang disimpan di slot replikasi logis dapat menyebabkan peningkatan penggunaan disk. Kami merekomendasikan agar Anda

mengatur ruang penggunaan alarm dalam instans PostgreSQL sumber ketika Anda menggunakan slot replikasi logis. Untuk informasi lebih lanjut tentang pengaturan atribut sambungan tambahan slotName, lihat [Pengaturan titik akhir dan Atribut Koneksi Ekstra \(ECA\) saat menggunakan PostgreSQL sebagai sumber DMS](#).

Prosedur berikut melalui pendekatan ini secara lebih rinci.

Menggunakan titik awal CDC asli untuk mengatur beban CDC dari titik akhir sumber PostgreSQL

1. Identifikasi slot replikasi logis yang digunakan oleh tugas replikasi sebelumnya (tugas induk) yang ingin Anda gunakan sebagai titik awal. Kemudian lakukan kueri tampilan `pg_replication_slots` pada basis data sumber Anda untuk memastikan bahwa slot ini tidak memiliki sambungan aktif. Jika ya, selesaikan dan tutup sebelum melanjutkan.

Untuk langkah-langkah berikut, anggaplah bahwa slot replikasi logis Anda `abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef`.

2. Buat titik akhir sumber baru yang mencakup pengaturan atribut sambungan tambahan berikut.

```
slotName=abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef;
```

3. Buat tugas baru khusus CDC menggunakan konsol, AWS CLI atau AWS DMS API. Misalnya, dengan menggunakan CLI Anda dapat menjalankan perintah `create-replication-task` berikut.

```
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:ZYX9WVUTSRQ0NM8LKJIHGF7ED6
--replication-instance-arn arn:aws:dms:us-west-2:012345678901:rep:AAAAAAAAAAAA5BB4CCC3DDDD2EE
--migration-type cdc --table-mappings "file://mappings.json" --cdc-start-position
"4AF/B00000D0"
--replication-task-settings "file://task-pg.json"
```

Dalam perintah sebelumnya, opsi berikut ditetapkan:

- Opsi `source-endpoint-arn` diatur menjadi nilai baru yang Anda buat di langkah 2.

- Opsi `replication-instance-arn` diatur menjadi nilai yang sama seperti untuk tugas induk dari langkah 1.
- Opsi `table-mappings` dan `replication-task-settings` ditetapkan menjadi nilai yang sama seperti untuk tugas induk dari langkah 1.
- Opsi `cdc-start-position` diatur menjadi nilai posisi awal. Untuk menemukan posisi awal ini, lakukan kueri tampilan `pg_replication_slots` pada basis data sumber Anda atau lihat detail konsol untuk tugas induk di langkah 1. Untuk informasi selengkapnya, lihat [Menentukan titik awal CDC asli](#).

Untuk mengaktifkan mode mulai CDC khusus saat membuat tugas khusus CDC baru menggunakan AWS DMS konsol, lakukan hal berikut:

- Di bagian Pengaturan tugas, untuk mode mulai CDC untuk transaksi sumber, pilih Aktifkan mode mulai CDC khusus.
- Untuk titik awal CDC khusus untuk transaksi sumber, pilih Tentukan nomor urutan log. Tentukan nomor perubahan sistem atau pilih Tentukan pos pemeriksaan pemulihan, dan berikan pos pemeriksaan Pemulihan.

Ketika tugas CDC ini berjalan, AWS DMS menimbulkan kesalahan jika slot replikasi logis yang ditentukan tidak ada. Hal ini juga menimbulkan kesalahan jika tugas tidak dibuat dengan pengaturan yang valid untuk `cdc-start-position`.

Ketika menggunakan titik awal CDC asli dengan plugin `pglogical` dan Anda ingin menggunakan slot replikasi baru, selesaikan langkah-langkah pengaturan berikut sebelum membuat tugas CDC.

Menggunakan slot replikasi baru yang sebelumnya tidak dibuat sebagai bagian dari tugas DMS lain

1. Buat slot replikasi, seperti yang ditunjukkan berikut:

```
SELECT * FROM pg_create_logical_replication_slot('replication_slot_name',  
'pglogical');
```

2. Setelah database membuat slot replikasi, dapatkan dan catat nilai `restart_lsn` dan `confirmed_flush_lsn` untuk slot:

```
select * from pg_replication_slots where slot_name like 'replication_slot_name';
```

Perhatikan bahwa posisi Mulai CDC Asli untuk tugas CDC yang dibuat setelah slot replikasi tidak boleh lebih tua dari nilai `confirmed_flush_lsn`.

[Untuk informasi tentang nilai `restart_lsn` dan `confirmed_flush_lsn`, lihat `pg_replication_slots`](#)

3. Buat node `pglogical`.

```
SELECT pglogical.create_node(node_name := 'node_name', dsn := 'your_dsn_name');
```

4. Buat dua set replikasi menggunakan `pglogical.create_replication_set` fungsi. Set replikasi pertama melacak pembaruan dan penghapusan untuk tabel yang memiliki kunci utama. Set replikasi kedua hanya menyisipkan, dan memiliki nama yang sama dengan set replikasi pertama, dengan awalan tambahan '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. Tambahkan tabel pada set replikasi.

```
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. Atur atribut sambungan tambahan (ECA) berikut ketika Anda membuat titik akhir sumber Anda.

```
PluginName=PGLOGICAL;slotName=slot_name;
```

Anda sekarang dapat membuat tugas CDC saja dengan titik awal asli PostgreSQL menggunakan slot replikasi baru. Untuk informasi lebih lanjut tentang plugin `pglogical`, lihat dokumentasi [pglogical](#) 3.7

Bermigrasi dari PostgreSQL ke PostgreSQL menggunakan AWS DMS

Ketika Anda bermigrasi dari mesin basis data selain PostgreSQL ke basis data PostgreSQL, AWS DMS hampir selalu merupakan alat migrasi terbaik untuk digunakan. Tapi ketika Anda bermigrasi dari basis data PostgreSQL ke basis data PostgreSQL, alat PostgreSQL bisa lebih efektif.

Menggunakan alat asli PostgreSQL untuk memigrasi data

Kami menyarankan Anda untuk menggunakan alat migrasi basis data seperti `pg_dump` dalam kondisi berikut:

- Anda memiliki migrasi yang homogen, yaitu Anda bermigrasi dari basis data PostgreSQL sumber ke basis data PostgreSQL target.
- Anda memigrasi seluruh basis data.
- Alat asli memungkinkan Anda untuk memigrasi data Anda dengan waktu downtime minimal.

Utilitas `pg_dump` menggunakan perintah `COPY` untuk membuat skema dan data dump dari basis data PostgreSQL. Skrip dump yang dibuat oleh `pg_dump` memuat data ke dalam basis data dengan nama yang sama dan membuat ulang tabel, indeks, dan kunci asing. Untuk memulihkan data ke basis data dengan nama yang berbeda, gunakan perintah `pg_restore` dan parameter `-d`.

Jika Anda memigrasi data dari basis data sumber PostgreSQL yang berjalan pada EC2 ke target Amazon RDS for PostgreSQL, Anda dapat menggunakan plugin `pglogical`.

Untuk informasi lebih lanjut tentang impor basis data PostgreSQL ke dalam Amazon RDS for PostgreSQL atau Amazon Aurora Edisi Kompatibel PostgreSQL, lihat <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/PostgreSQL.Procedural.Importing.html>.

Menggunakan DMS untuk memigrasi data dari PostgreSQL ke PostgreSQL

AWS DMS dapat memigrasi data, misalnya, dari sumber basis data PostgreSQL yang ada di lokasi ke instans target Amazon RDS for PostgreSQL atau Aurora PostgreSQL. Jenis data PostgreSQL inti atau basic paling sering bermigrasi dengan berhasil.

Note

Ketika mereplikasi tabel dipartisi dari sumber PostgreSQL ke target PostgreSQL, Anda tidak perlu menyebutkan tabel induk sebagai bagian dari kriteria seleksi dalam tugas DMS. Menyebutkan tabel induk menyebabkan data diduplikasi dalam tabel anak pada target, dapat

menyebabkan pelanggaran PK. Dengan memilih tabel anak saja dalam kriteria pemilihan pemetaan tabel, tabel induk secara otomatis diisi.

Jenis data yang didukung pada basis data sumber tetapi tidak didukung pada target mungkin tidak bermigrasi dengan berhasil. AWS DMS mengalirkan beberapa jenis data sebagai string jika jenis data tidak diketahui. Beberapa jenis data, seperti XML dan JSON, dapat berhasil bermigrasi sebagai file kecil tetapi dapat gagal jika mereka merupakan dokumen besar.

Saat melakukan migrasi jenis data, perhatikan hal-hal berikut:

- Dalam beberapa kasus, jenis data PostgreSQL NUMERIC(p,s) tidak menentukan presisi dan skala apa pun. Untuk DMS versi 3.4.2 dan versi sebelumnya, DMS menggunakan presisi 28 dan skala 6 secara default, NUMERIC (28,6). Misalnya, nilai 0.611111104488373 dari sumber dikonversi menjadi 0.611111 pada target PostgreSQL.
- Tabel dengan jenis data ARRAY harus memiliki kunci primer. Tabel dengan jenis data ARRAY yang kekurangan kunci primer akan ditangguhkan selama beban penuh.

Tabel berikut menunjukkan sumber jenis data PostgreSQL dan apakah mereka dapat berhasil dimigrasi.

Jenis data	Berhasil bermigrasi	Bermigrasi sebagian	Tidak bermigrasi	Komentar
INTEGER	X			
SMALLINT	X			
BIGINT	X			
NUMERIC/DECIMAL(p,s)		X		Di mana $0 < p < 39$ dan $0 < s$
NUMERIC/DECIMAL		X		Di mana $p > 38$ atau $p = s = 0$

Jenis data	Berhasil bermigrasi	Bermigrasi sebagian	Tidak bermigrasi	Komentar
REAL	X			
DOUBLE	X			
SMALLSERIAL	X			
SERIAL	X			
BIGSERIAL	X			
MONEY	X			
CHAR		X		Tanpa presisi tertentu
CHAR(N)	X			
VARCHAR		X		Tanpa presisi tertentu
VARCHAR(N)	X			
TEXT	X			
BYTEA	X			

Jenis data	Berhasil bermigrasi	Bermigrasi sebagian	Tidak bermigrasi	Komentar
TIMESTAMP	X			Nilai tak terhingga positif dan negatif dipotong menjadi '9999-12-31 23:59:59 'dan '4713-01-01 00:00:00 BC' masing-masing.
TIMESTAMP WITH TIME ZONE		X		
DATE	X			
TIME	X			
TIME WITH TIME ZONE		X		
INTERVAL		X		
BOOLEAN	X			
ENUM			X	
CIDR	X			
INET			X	
MACADDR			X	
TSVECTOR			X	
TSQUERY			X	

Jenis data	Berhasil bermigrasi	Bermigrasi sebagian	Tidak bermigrasi	Komentar
XML		X		
POINT	X			Jenis data spasial PostGIS
LINE			X	
LSEG			X	
BOX			X	
PATH			X	
POLYGON	X			Jenis data spasial PostGIS
CIRCLE			X	
JSON		X		
ARRAY	X			Memerlukan Kunci Primer
COMPOSITE			X	
RANGE			X	
LINestring	X			Jenis data spasial PostGIS
MULTIPOINT	X			Jenis data spasial PostGIS

Jenis data	Berhasil bermigrasi	Bermigrasi sebagian	Tidak bermigrasi	Komentar
MULTILINESTRING	X			Jenis data spasial PostGIS
MULTIPOLYGON	X			Jenis data spasial PostGIS
GEOMETRYCOLLECTION	X			Jenis data spasial PostGIS

Memigrasikan jenis data spasial PostGIS

Data spasial mengidentifikasi informasi geometri dari suatu objek atau lokasi di ruang. Basis data relasional objek PostgreSQL mendukung jenis data spasial PostGIS.

Sebelum memigrasi objek data spasial PostgreSQL, pastikan plugin PostGIS diaktifkan pada tingkat global. Melakukannya dapat memastikan bahwa AWS DMS menciptakan kolom data spasial sumber yang tepat untuk instans DB target PostgreSQL.

Untuk migrasi homogen PostgreSQL ke PostgreSQL, AWS DMS mendukung migrasi jenis dan subjenis objek data geometris dan geografis (koordinat geodesi) PostGIS seperti berikut:

- POINT
- LINESTRING
- POLYGON
- MULTIPOINT
- MULTILINESTRING
- MULTIPOLYGON
- GEOMETRYCOLLECTION

Menghapus artefak AWS DMS dari basis data sumber PostgreSQL

Untuk menangkap peristiwa DDL, AWS DMS membuat berbagai artefak dalam basis data PostgreSQL ketika tugas migrasi dimulai. Saat tugas selesai, Anda mungkin ingin menghapus artefak ini.

Untuk menghapus artefak, keluarkan pernyataan berikut (dalam urutan saat muncul), di mana {AmazonRDSMigration} adalah skema di mana artefak dibuat. Menjatuhkan skema harus dilakukan dengan sangat hati-hati. Jangan pernah menjatuhkan skema operasional, apalagi skema publik.

```
drop event trigger awsdms_intercept_ddl;
```

Pemicu peristiwa bukan milik skema tertentu.

```
drop function {AmazonRDSMigration}.awsdms_intercept_ddl()  
drop table {AmazonRDSMigration}.awsdms_ddl_audit  
drop schema {AmazonRDSMigration}
```

Pengaturan konfigurasi tambahan ketika menggunakan basis data PostgreSQL sebagai sumber DMS

Anda dapat menambahkan pengaturan konfigurasi tambahan ketika memigrasi data dari basis data PostgreSQL dengan dua cara:

- Anda dapat menambahkan nilai ke atribut sambungan tambahan untuk menangkap peristiwa DDL dan menentukan skema di mana artefak basis data DL operasional dibuat. Untuk informasi selengkapnya, lihat [Pengaturan titik akhir dan Atribut Koneksi Ekstra \(ECA\) saat menggunakan PostgreSQL sebagai sumber DMS](#).
- Anda dapat mengganti parameter string sambungan. Pilih opsi ini untuk melakukan salah satu dari berikut:
 - Tentukan internal parameter AWS DMS. Parameter tersebut jarang diperlukan sehingga tidak tampak di antarmuka pengguna.
 - Tentukan nilai pass-through (passthru) untuk klien basis data tertentu. AWS DMS memuat parameter pass-through dalam string sambungan yang diberikan pada klien basis data.
- Dengan menggunakan parameter tingkat tabel di REPLICA IDENTITY PostgreSQL versi 9.4 dan yang lebih tinggi, Anda dapat mengontrol informasi yang ditulis ke log write-ahead (WALS). Secara spesifik, hal itu dilakukan untuk WALS yang mengidentifikasi baris yang diperbarui atau

dihapus. `REPLICA IDENTITY FULL` mencatat nilai-nilai lama dari semua kolom di baris. Gunakan `REPLICA IDENTITY FULL` dengan hati-hati untuk setiap table seraya `FULL` menghasilkan sejumlah tambahan WALs yang mungkin tidak diperlukan. Untuk informasi lebih lanjut, lihat [ALTER TABLE-REPLICA IDENTITY](#)

Menggunakan pengaturan titik akhir MapBooleanAsBoolean PostgreSQL

Anda dapat menggunakan pengaturan titik akhir PostgreSQL untuk memetakan boolean sebagai boolean dari sumber PostgreSQL Anda ke target Amazon Redshift. Secara default, tipe `BOOLEAN` dimigrasikan sebagai `varchar (5)`. Anda dapat menentukan `MapBooleanAsBoolean` untuk membiarkan PostgreSQL memigrasikan jenis boolean sebagai boolean seperti yang ditunjukkan pada contoh berikut.

```
--postgre-sql-settings '{"MapBooleanAsBoolean": true}'
```

Perhatikan bahwa Anda harus mengatur pengaturan ini pada titik akhir sumber dan target agar dapat diterapkan.

Karena MySQL tidak memiliki tipe `BOOLEAN`, gunakan aturan transformasi daripada pengaturan ini saat memigrasi data `BOOLEAN` ke MySQL.

Pengaturan titik akhir dan Atribut Koneksi Ekstra (ECA) saat menggunakan PostgreSQL sebagai sumber DMS

Anda dapat menggunakan pengaturan titik akhir dan atribut koneksi tambahan (ECA) untuk mengonfigurasi basis data sumber PostgreSQL Anda. Anda menentukan pengaturan titik akhir saat membuat titik akhir sumber menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--postgre-sql-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan endpoint dan ECA yang dapat Anda gunakan dengan PostgreSQL sebagai sumber.

Nama atribut	Deskripsi
CaptureDDLs	Untuk menangkap peristiwa DDL, AWS DMS menciptakan berbagai artefak dalam basis data PostgreSQL ketika tugas dimulai. Anda nantinya dapat menghapus artefak

Nama atribut	Deskripsi
	<p>ini seperti yang dijelaskan di Menghapus artefak AWS DMS dari basis data sumber PostgreSQL.</p> <p>Jika nilai ini disetel ke <code>false</code>, Anda tidak perlu membuat tabel atau pemicu pada database sumber.</p> <p>Acara DDL yang dikeluarkan ditangkap.</p> <p>Nilai default: <code>true</code></p> <p>Nilai yang benar: benar/salah</p> <p>Contoh: <code>--postgre-sql-settings '{"CaptureDDLs": true}'</code></p>
ConsumeMonotonicEvents	<p>Digunakan untuk mengontrol bagaimana transaksi monolitik dengan Log Sequence Numbers (LSNs) duplikat direplikasi. Bila parameter ini <code>false</code>, peristiwa dengan LSNs duplikat dikonsumsi dan direplikasi pada target. Bila parameter ini <code>true</code>, hanya peristiwa pertama yang direplikasi sementara peristiwa dengan LSNs duplikat tidak dikonsumsi atau direplikasi pada target.</p> <p>Nilai default: <code>false</code></p> <p>Nilai yang benar: salah/benar</p> <p>Contoh: <code>--postgre-sql-settings '{"ConsumeMonotonicEvents": true}'</code></p>
DdlArtifactsSchema	<p>Menetapkan skema di mana artefak basis data DL operasional dibuat.</p> <p>Nilai default: publik</p> <p>Nilai yang benar: String</p> <p>Contoh: <code>--postgre-sql-settings '{"DdlArtifactsSchema": "xyzddlSchema"}</code></p>

Nama atribut	Deskripsi
ExecuteTimeout	<p>Menetapkan pernyataan timeout klien untuk instans PostgreSQL, dalam hitungan detik. Nilai default adalah 60 detik.</p> <p>Contoh: <code>--postgres-sql-settings '{"ExecuteTimeout": 100}'</code></p>
FailTasksOnLobTruncation	<p>Ketika diatur menjadi <code>true</code>, nilai ini menyebabkan tugas gagal jika ukuran kolom LOB sebenarnya lebih besar dari yang <code>LobMaxSize</code> tentukan.</p> <p>Jika tugas diatur ke modus LOB terbatas dan opsi ini diatur menjadi benar, tugas gagal alih-alih memotong data LOB.</p> <p>Nilai default: salah</p> <p>Nilai yang benar: Boolean</p> <p>Contoh: <code>--postgres-sql-settings '{"FailTasksOnLobTruncation": true}'</code></p>
fetchCacheSize	<p>Atribut koneksi tambahan (ECA) ini menetapkan jumlah baris yang akan diambil kursor selama operasi beban penuh. Bergantung pada sumber daya yang tersedia dalam contoh replikasi, Anda dapat menyesuaikan nilainya lebih tinggi atau lebih rendah.</p> <p>Nilai default: 10000</p> <p>Nilai yang benar: Angka</p> <p>Contoh ECA: <code>fetchCacheSize=10000;</code></p>

Nama atribut	Deskripsi
HeartbeatFrequency	<p>Menetapkan frekuensi WAL heartbeat (dalam menit).</p> <p>Nilai default: 5</p> <p>Nilai yang benar: Angka</p> <p>Contoh: <code>--postgresql-settings '{"HeartbeatFrequency": 1}'</code></p>
HeartbeatSchema	<p>Menetapkan skema di mana artefak heartbeat dibuat.</p> <p>Nilai default: <code>public</code></p> <p>Nilai yang benar: String</p> <p>Contoh: <code>--postgresql-settings '{"HeartbeatSchema": "xyzheartbeatSchema"}'</code></p>
MapJsonbAsClob	<p>Secara default, AWS DMS memetakan JSONB ke NCLOB. Anda dapat menentukan MapJsonbAsClob untuk membiarkan PostgreSQL memigrasikan tipe JSONB sebagai CLOB.</p> <p>Contoh: <code>--postgresql-settings='{"MapJsonbAsClob": "true"}'</code></p>

Nama atribut	Deskripsi
MapLongVarcharAs	<p>Secara default, AWS DMS memetakan VARCHAR ke WSTRING. Anda dapat menentukan MapLongVarcharAs untuk membiarkan PostgreSQL memigrasikan tipe VARCHAR (N) (di mana N lebih besar dari 16387) ke jenis berikut:</p> <ul style="list-style-type: none">• WSTRING• CLOB• NCLOB <p>Contoh: <code>--postgresql-settings='{ "MapLongVarcharAs": "CLOB" }'</code></p>

Nama atribut	Deskripsi
MapUnboundedNumericAsString	<p>Parameter ini memperlakukan kolom dengan jenis data NUMERIC tak terbatas sebagai STRING agar berhasil bermigrasi tanpa kehilangan presisi dari nilai numerik. Gunakan parameter ini hanya untuk replikasi dari sumber PostgreSQL ke target PostgreSQL, atau basis data dengan kompatibilitas PostgreSQL.</p> <p>Nilai default: <code>false</code></p> <p>Nilai valid: <code>false/true</code></p> <p>Contoh: <code>--postgre-sql-settings '{"MapUnboundedNumericAsString": true}'</code></p> <p>Menggunakan parameter ini dapat mengakibatkan beberapa degradasi performa replikasi karena transformasi dari numerik ke string dan kembali ke numerik. Parameter ini didukung untuk digunakan oleh DMS versi 3.4.4 dan versi yang lebih tinggi</p> <div data-bbox="688 1100 1507 1753" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Hanya gunakan MapUnboundedNumericAsString di sumber PostgreSQL dan titik akhir target bersama-sama. Penggunaan titik akhir PostgreSQL sumber membatasi presisi hingga 28 selama CDC. MapUnboundedNumericAsString Penggunaan MapUnboundedNumericAsString pada titik akhir target, memigrasi data dengan Precision 28 Scale 6. Jangan gunakan MapUnboundedNumericAsString dengan target non-PostgreSQL.</p></div>

Nama atribut	Deskripsi
PluginName	<p>Menentukan plugin yang digunakan untuk membuat slot replikasi.</p> <p>Nilai yang valid: <code>pglogical</code> , <code>test_decoding</code></p> <p>Contoh: <code>--postgre-sql-settings '{"Plugin Name": "test_decoding"}'</code></p>

Nama atribut	Deskripsi
SlotName	<p>Menetapkan nama slot replikasi logis yang dibuat sebelumnya untuk beban CDC dari instans sumber PostgreSQL.</p> <p>Ketika digunakan dengan parameter permintaan API <code>CdcStartPosition</code> AWS DMS, atribut ini juga memungkinkan menggunakan titik awal CDC asli. DMS melakukan verifikasi bahwa slot replikasi logis tertentu ada sebelum memulai tugas beban CDC. Hal ini juga memverifikasi bahwa tugas dibuat dengan pengaturan <code>CdcStartPosition</code> yang valid. Jika slot yang ditentukan tidak ada atau tugas tidak memiliki pengaturan <code>CdcStartPosition</code> yang valid, DMS memunculkan kesalahan.</p> <p>Untuk informasi selengkapnya tentang pengaturan parameter permintaan <code>CdcStartPosition</code>, lihat Menentukan titik awal CDC asli. Untuk informasi lebih lanjut tentang penggunaan <code>CdcStartPosition</code>, lihat dokumentasi untuk operasi API <code>CreateReplicationTask</code>, <code>StartReplicationTask</code>, dan <code>ModifyReplicationTask</code> di AWS Database Migration Service Referensi API.</p> <p>Nilai yang benar: String</p> <p>Contoh: <code>--postgresql-settings '{"SlotName": "abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef"}'</code></p>
unboundedVarcharMaxSize	<p>Atribut Koneksi Ekstra (ECA) ini mendefinisikan ukuran maksimum kolom data yang didefinisikan sebagai tipe <code>VarChar</code> tanpa penentu panjang maksimum. Defaultnya adalah 8000 byte. Nilai maksimum adalah 10485760 byte.</p>

Keterbatasan menggunakan basis data PostgreSQL sebagai sumber DMS

Keterbatasan berikut berlaku ketika menggunakan PostgreSQL sebagai sumber untuk AWS DMS:

- AWS DMS tidak bekerja dengan Amazon RDS for PostgreSQL 10.4 atau Amazon Aurora PostgreSQL 10.4 baik sebagai sumber maupun target.
- Tabel yang ditangkap harus memiliki kunci primer. Jika tabel tidak memiliki kunci primer, AWS DMS mengabaikan operasi catatan DELETE dan UPDATE untuk tabel tersebut. Sebagai solusinya, lihat [Mengaktifkan pengambilan data perubahan \(CDC\)](#) menggunakan replikasi logis.

Catatan: Kami tidak menyarankan migrasi tanpa Kunci Primer/Indeks Unik, jika tidak, batasan tambahan berlaku seperti kemampuan penerapan Batch “TIDAK”, kemampuan LOB Penuh, Validasi Data, dan ketidakmampuan untuk mereplikasi ke target Redshift secara efisien.

- AWS DMS mengabaikan upaya untuk memperbarui segmen kunci primer. Dalam kasus ini, target mengidentifikasi pembaruan sebagai salah satu yang tidak memperbarui baris apapun. Namun, karena hasil memperbarui kunci primer di PostgreSQL tidak dapat diprediksi, tidak ada catatan yang ditulis ke tabel pengecualian.
- AWS DMS tidak mendukung opsi Mulai Proses Perubahan dari Timestamp.
- AWS DMS tidak mereplikasi perubahan yang dihasilkan dari operasi partisi atau subpartisi (ADD, DROP, atau TRUNCATE).
- Replikasi beberapa tabel dengan nama yang sama di mana setiap nama memiliki kasus yang berbeda (misalnya, tabel1, TABEL1, dan Tabel1) dapat menyebabkan perilaku tak terduga. Karena masalah ini, AWS DMS tidak mendukung jenis replikasi ini.
- Dalam kebanyakan kasus, AWS DMS mendukung pengolahan perubahan pernyataan CREATE, ALTER, dan DROP DDL untuk tabel. AWS DMS tidak mendukung pemrosesan perubahan ini jika tabel diadakan dalam fungsi dalam atau prosedur blok tubuh atau dalam konstruksi bersarang lainnya.

Sebagai contoh, perubahan berikut tidak ditangkap.

```
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;
```

```
$$;
```

- Saat ini, jenis data boolean dalam sumber PostgreSQL dimigrasi ke target SQL Server sebagai jenis data bit dengan nilai-nilai yang tidak konsisten. Sebagai solusinya, pra-buat tabel dengan tipe VARCHAR(1) data untuk kolom (atau minta AWS DMS membuat tabel). Kemudian memiliki pemrosesan hilir memperlakukan “F” sebagai False (Salah) dan “T” sebagai True (Benar).
- AWS DMS tidak mendukung pemrosesan perubahan operasi TRUNCATE.
- Jenis data OID LOB tidak bermigrasi ke target.
- AWS DMS mendukung tipe data PostGIS hanya untuk migrasi homogen.
- Jika sumber Anda adalah basis data PostgreSQL yang lokal atau pada instans Amazon EC2, pastikan bahwa plugin output test_decoding diinstal pada titik akhir sumber Anda. Anda dapat menemukan plugin ini di paket kontrib PostgreSQL. Untuk informasi lebih lanjut tentang plugin test_decoding, lihat [Dokumentasi PostgreSQL](#).
- AWS DMS tidak mendukung perubahan pemrosesan untuk mengatur dan tidak mengatur kolom nilai default (menggunakan klausa ALTER COLUMN SET DEFAULT pada pernyataan ALTER TABLE).
- AWS DMS tidak mendukung pemrosesan pengubahan untuk mengatur kolom nullabilitas (menggunakan klausa ALTER COLUMN[SET|DROP] NOT NULL pada pernyataan ALTER TABLE).
- Ketika replikasi logis diaktifkan, jumlah maksimum perubahan yang disimpan dalam memori per transaksi adalah 4 MB. Setelah itu, perubahan dijatuhkan ke disk. Hasilnya ReplicationSlotDiskUsage meningkat, dan restart_lsn tidak maju sampai transaksi selesai atau berhenti dan rollback selesai. Karena merupakan transaksi yang panjang, itu dapat memakan waktu lama untuk rollback. Jadi, hindari transaksi yang berjalan lama atau banyak sub-transaksi saat replikasi logis diaktifkan. Sebaliknya, pisah transaksi menjadi beberapa transaksi yang lebih kecil.

Pada Aurora PostgreSQL versi 13 dan yang lebih baru, Anda dapat menyetel `logical_decoding_work_mem` parameter untuk mengontrol ketika DMS tumpah mengubah data ke disk. Untuk informasi selengkapnya, lihat [Menumpahkan file di Aurora PostgreSQL](#).

- Tabel dengan jenis data ARRAY harus memiliki kunci primer. Tabel dengan jenis data ARRAY tanpa kunci primer akan ditangguhkan selama beban penuh.
- AWS DMS tidak mendukung replikasi tabel dipartisi. Ketika tabel dipartisi terdeteksi, hal berikut terjadi:
 - Titik akhir melaporkan daftar tabel induk dan anak.

- AWS DMS menciptakan tabel pada target sebagai tabel biasa dengan sifat yang sama seperti tabel yang dipilih.
- Jika tabel induk dalam basis data sumber memiliki nilai kunci primer yang sama dengan tabel anaknya, kesalahan “duplikat kunci” muncul.
- Untuk mereplikasi tabel dipartisi dari sumber PostgreSQL ke target PostgreSQL, pertama-tama secara manual buatlah tabel induk dan anak pada target. Kemudian tentukan tugas terpisah untuk mereplikasi ke tabel tersebut. Dalam kasus seperti itu, atur konfigurasi tugas Potong sebelum memuat.
- Jenis data NUMERIC PostgreSQL tidak tentu dalam hal ukuran. Saat mentransfer data yang merupakan jenis data NUMERIC tetapi tanpa presisi dan skala, DMS menggunakan NUMERIC(28,6) (presisi 28 dan skala 6) secara default. Sebagai contoh, nilai 0.6111111104488373 dari sumber dikonversi menjadi 0.611111 pada target PostgreSQL.
- AWS DMS mendukung Aurora PostgreSQL Serverless V1 sebagai sumber untuk tugas beban penuh saja. AWS DMS mendukung Aurora PostgreSQL Serverless V2 sebagai sumber untuk beban penuh, beban penuh dan CDC, dan tugas khusus CDC.
- AWS DMS tidak mendukung replikasi tabel dengan indeks unik yang dibuat dengan fungsi gabungan.
- Saat menggunakan mode LOB, tabel sumber dan tabel target yang sesuai harus memiliki Kunci Utama yang identik. Jika salah satu tabel tidak memiliki Kunci Utama, hasil operasi catatan DELETE dan UPDATE tidak akan dapat diprediksi.
- Saat menggunakan fitur Parallel Load, segmentasi tabel menurut partisi atau sub-partisi tidak didukung. Untuk informasi selengkapnya tentang Beban Paralel, lihat [Menggunakan beban paralel untuk tabel, tampilan, dan koleksi yang dipilih](#)
- AWS DMS tidak mendukung Kendala yang Ditangguhkan.
- AWS DMS versi 3.4.7 mendukung PostgreSQL 14.x sebagai sumber dengan keterbatasan ini:
 - AWS DMS tidak mendukung pemrosesan perubahan komit dua fase.
 - AWS DMS tidak mendukung replikasi logis untuk mengalirkan transaksi yang sedang berlangsung lama.
- AWS DMS tidak mendukung CDC untuk Amazon RDS Proxy untuk PostgreSQL sebagai sumber.
- Saat menggunakan [filter sumber](#) yang tidak berisi kolom Kunci Utama, DELETE operasi tidak akan ditangkap.
- Jika database sumber Anda juga merupakan target untuk sistem replikasi pihak ketiga lainnya, perubahan DDL mungkin tidak bermigrasi selama CDC. Karena situasi itu dapat mencegah pemicu

`awsdms_intercept_ddl` peristiwa menembak. Untuk mengatasi situasi tersebut, ubah pemicu itu pada basis data sumber Anda sebagai berikut:

```
alter event trigger awsdms_intercept_ddl enable always;
```

- AWS DMS tidak mendukung CDC untuk cluster database Multi-AZ Amazon RDS untuk PostgreSQL sebagai sumber, karena RDS untuk cluster database PostgreSQL Multi-AZ tidak mendukung replikasi logis.

Jenis data sumber untuk PostgreSQL

Tabel berikut menunjukkan jenis data sumber PostgreSQL yang didukung saat menggunakan AWS DMS dan pemetaan default untuk jenis data AWS DMS.

Untuk informasi tentang cara melihat jenis data yang dipetakan dalam target, lihat bagian titik akhir target yang Anda gunakan.

Untuk informasi tambahan tentang jenis data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Jenis data PostgreSQL	Jenis data DMS
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
NOMERIC (p,s)	Jika presisi adalah dari 0 sampai 38, maka gunakan NUMERIC. Jika presisi adalah 39 atau lebih besar, maka gunakan STRING.
DECIMAL(P,S)	Jika presisi adalah dari 0 sampai 38, maka gunakan NUMERIC. Jika presisi adalah 39 atau lebih besar, maka gunakan STRING.

Jenis data PostgreSQL	Jenis data DMS
REAL	REAL4
DOUBLE	REAL8
SMALLSERIAL	INT2
SERIAL	INT4
BIGSERIAL	INT8
MONEY	NUMERIC(38,4) Jenis data MONEY dipetakan ke FLOAT di SQL Server.
CHAR	WSTRING (1)
CHAR(N)	WSTRING (n)
VARCHAR(N)	WSTRING (n)
TEXT	NCLOB
BYTEA	BLOB
TIMESTAMP	DATETIME
TIMESTAMP DENGAN ZONA WAKTU	DATETIME
DATE	DATE
TIME	TIME
TIME WITH TIME ZONE	TIME
INTERVAL	STRING (128)—1 TAHUN, 2 BULAN, 3 HARI, 4 JAM, 5 MENIT, 6 DETIK
BOOLEAN	CHAR (5) salah atau benar

Jenis data PostgreSQL	Jenis data DMS
ENUM	STRING (64)
CIDR	STRING (50)
INET	STRING (50)
MACADDR	STRING (18)
BIT (n)	STRING (n)
BIT VARYING	STRING (n)
UUID	STRING
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))"
PATH	CLOB "((x1,y1),(xn,yn))"
POLYGON	CLOB "((x1,y1),(xn,yn))"
CIRCLE	STRING (255) "(x,y,r)"
JSON	NCLOB
JSONB	NCLOB
ARRAY	NCLOB

Jenis data PostgreSQL	Jenis data DMS
COMPOSITE	NCLOB
HSTORE	NCLOB
INT4RANGE	STRING (255)
INT8RANGE	STRING (255)
NUMRANGE	STRING (255)
STRRANGE	STRING (255)

Menggunakan jenis data sumber LOB untuk PostgreSQL

Ukuran kolom PostgreSQL memengaruhi konversi jenis data LOB PostgreSQL ke jenis data AWS DMS. Untuk menggunakannya, ambil langkah-langkah berikut untuk jenis data AWS DMS berikut:

- BLOB - Atur Batas ukuran LOB hingga nilai Ukuran LOB maksimum (KB) pada saat pembuatan tugas.
- CLOB - Replikasi menangani setiap karakter sebagai karakter UTF8. Oleh karena itu, temukan panjang teks karakter terpanjang di kolom, yang ditunjukkan di sini sebagai `max_num_chars_text`. Gunakan panjang ini untuk menentukan nilai untuk Batasi ukuran LOB ke. Jika data memuat karakter 4-byte, kalikan dengan 2 untuk menentukan nilai Batas ukuran LOB hingga, yang dalam bentuk byte. Dalam kasus ini, Batas ukuran LOB hingga sama dengan `max_num_chars_text` dikalikan dengan 2.
- NCLOB - Replikasi menangani setiap karakter sebagai karakter double-byte. Oleh karena itu, cari panjang teks karakter terpanjang di kolom (`max_num_chars_text`) dan kalikan dengan 2. Anda melakukan ini untuk menentukan nilai untuk Batasi ukuran LOB ke. Dalam kasus ini, Batas ukuran LOB hingga sama dengan `max_num_chars_text` dikalikan dengan 2. Jika data memuat karakter 4-byte, kalikan dengan 2 lagi. Dalam kasus ini, Batas ukuran LOB hingga sama dengan `max_num_chars_text` dikalikan dengan 4.

Menggunakan basis data yang kompatibel dengan MySQL sebagai sumber untuk AWS DMS

Anda dapat memigrasikan data dari database yang kompatibel dengan MySQL (MySQL, MariaDB, atau Amazon Aurora MySQL) menggunakan Database Migration Service. AWS

Untuk informasi tentang versi MySQL AWS DMS yang mendukung sebagai sumber, lihat [Sumber untuk AWS DMS](#)

Anda dapat menggunakan SSL untuk mengenkripsi koneksi antara titik akhir yang kompatibel dengan MySQL dan instans replikasi. Untuk informasi lebih lanjut tentang penggunaan SSL dengan titik akhir yang kompatibel dengan MySQL, lihat [Menggunakan SSL dengan AWS Database Migration Service](#).

Di bagian berikut, syarat “dikelola sendiri” berlaku untuk setiap basis data yang diinstal baik lokal atau di Amazon EC2. Syarat "terkelola AWS" berlaku untuk basis data apa pun di Amazon RDS, Amazon Aurora, atau Amazon S3.

Untuk detail tambahan tentang bekerja dengan database yang kompatibel dengan MySQL dan AWS DMS, lihat bagian berikut.

Topik

- [Migrasi dari MySQL ke MySQL menggunakan AWS DMS](#)
- [Menggunakan database yang kompatibel dengan MySQL sebagai sumber AWS DMS](#)
- [Menggunakan database yang kompatibel dengan MySQL yang dikelola sendiri sebagai sumber AWS DMS](#)
- [Menggunakan database yang kompatibel dengan MySQL AWS-managed sebagai sumber untuk AWS DMS](#)
- [Batasan dalam menggunakan database MySQL sebagai sumber AWS DMS](#)
- [Support untuk transaksi XA](#)
- [Pengaturan titik akhir saat menggunakan MySQL sebagai sumber AWS DMS](#)
- [Jenis data sumber untuk MySQL](#)

Migrasi dari MySQL ke MySQL menggunakan AWS DMS

Untuk migrasi heterogen, tempat Anda bermigrasi dari mesin database selain MySQL ke database MySQL AWS DMS, hampir selalu merupakan alat migrasi terbaik untuk digunakan. Tetapi untuk

migrasi homogen, di mana Anda bermigrasi dari database MySQL ke database MySQL, kami sarankan Anda menggunakan proyek migrasi migrasi data homogen. migrasi data homogen menggunakan alat database asli untuk memberikan kinerja dan akurasi migrasi data yang ditingkatkan jika dibandingkan dengan. AWS DMS

Menggunakan database yang kompatibel dengan MySQL sebagai sumber AWS DMS

Sebelum Anda mulai bekerja dengan database MySQL sebagai sumber AWS DMS untuk, pastikan bahwa Anda memiliki prasyarat berikut. Prasyarat ini berlaku untuk sumber yang dikelola sendiri atau yang dikelola sendiri. AWS

Anda harus memiliki akun AWS DMS yang memiliki peran Admin Replikasi. Peran itu memerlukan keistimewaan berikut:

- REPLICATION CLIENT — Hak istimewa ini diperlukan untuk tugas CDC saja. Dengan kata lain, full-load-only tugas tidak memerlukan hak istimewa ini.
- REPLICATION SLAVE — Hak istimewa ini diperlukan untuk tugas CDC saja. Dengan kata lain, full-load-only tugas tidak memerlukan hak istimewa ini.
- SUPER — Hak istimewa ini diperlukan hanya dalam versi MySQL sebelum 5.6.6.

AWS DMS Pengguna juga harus memiliki hak SELECT untuk tabel sumber yang ditunjuk untuk replikasi.

Menggunakan database yang kompatibel dengan MySQL yang dikelola sendiri sebagai sumber AWS DMS

Anda dapat menggunakan basis data yang kompatibel dengan MySQL yang dikelola sendiri berikut sebagai sumber untuk 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

Untuk menggunakan CDC, pastikan untuk mengaktifkan binary logging. Untuk mengaktifkan binary logging, parameter berikut harus dikonfigurasi di MySQL `my.ini` (Windows) atau file `my.cnf` (UNIX).

Parameter	Nilai
<code>server_id</code>	Atur parameter ini supaya nilainya 1 atau lebih besar.
<code>log-bin</code>	Atur jalur ke berkas log biner, seperti <code>log-bin=E:\MySQL_Logs\BinLog</code> . Jangan sertakan ekstensi file.
<code>binlog_format</code>	Atur parameter ini menjadi ROW. Kami merekomendasikan pengaturan ini selama replikasi karena dalam kasus-kasus tertentu ketika <code>binlog_format</code> diatur menjadi STATEMENT , inkonsistensi dapat terjadi ketika mereplikasi data ke target. Mesin basis data juga menulis data yang tidak konsisten yang mirip dengan target ketika <code>binlog_format</code> diatur menjadi MIXED, karena mesin basis data secara otomatis beralih ke logging berbasis STATEMENT yang dapat mengakibatkan penulisan data yang tidak konsisten pada basis data target.
<code>expire_logs_days</code>	Atur parameter ini supaya nilainya 1 atau lebih besar. Untuk mencegah penggunaan ruang disk secara berlebihan, kami sarankan Anda tidak menggunakan nilai default 0.
<code>binlog_checksum</code>	Setel parameter ini NONE untuk DMS versi 3.4.7 atau sebelumnya.
<code>binlog_row_image</code>	Atur parameter ini menjadi FULL.
<code>log_slave_updates</code>	Atur parameter ini menjadi TRUE jika Anda menggunakan MySQL atau MariaDB replika baca sebagai sumber.

Jika sumber Anda menggunakan mesin basis data NDB (clustered), parameter berikut harus dikonfigurasi untuk mengaktifkan CDC pada tabel yang menggunakan mesin penyimpanan. Tambahkan perubahan ini di MySQL `my.ini` (Windows) atau file `my.cnf` (UNIX).

Parameter	Nilai
<code>ndb_log_bin</code>	Atur parameter ini menjadi ON. Nilai ini memastikan bahwa perubahan dalam clustered tables tercatat di log biner.
<code>ndb_log_u</code> <code>pdate_as_write</code>	Atur parameter ini menjadi OFF. Nilai ini mencegah penulisan pernyataan UPDATE sebagai pernyataan INSERT dalam log biner.
<code>ndb_log_u</code> <code>pdated_only</code>	Atur parameter ini menjadi OFF. Nilai ini memastikan bahwa log biner berisi seluruh baris dan bukan hanya kolom yang berubah.

Menggunakan database yang kompatibel dengan MySQL AWS-managed sebagai sumber untuk AWS DMS

Anda dapat menggunakan database yang kompatibel dengan MySQL AWS-managed berikut sebagai sumber untuk: AWS DMS

- MySQL Community Edition
- MariaDB Community Edition
- Edisi yang Kompatibel dengan Amazon Aurora MySQL

Saat menggunakan database yang kompatibel dengan MySQL AWS-managed sebagai sumber AWS DMS, pastikan Anda memiliki prasyarat berikut untuk CDC:

- Untuk mengaktifkan log biner untuk RDS untuk MySQL dan untuk RDS untuk MariaDB, aktifkan backup otomatis pada tingkat instans. Untuk mengaktifkan log biner untuk cluster MySQL Aurora, ubah variabel dalam grup parameter. `binlog_format`

Untuk informasi selengkapnya tentang menyiapkan pencadangan otomatis, lihat [Bekerja dengan pencadangan otomatis di Panduan Pengguna](#) Amazon RDS.

Untuk informasi selengkapnya tentang menyiapkan pencatatan biner untuk database Amazon RDS for MySQL, [lihat Menyetel format logging biner di Panduan](#) Pengguna Amazon RDS.

Untuk informasi selengkapnya tentang menyiapkan pencatatan biner untuk klaster MySQL Aurora, lihat [Bagaimana cara mengaktifkan pencatatan biner untuk klaster MySQL Amazon Aurora saya?](#) .

- Jika Anda berencana untuk menggunakan CDC, aktifkan logging biner. Untuk informasi selengkapnya tentang menyiapkan pencatatan biner untuk database Amazon RDS for MySQL, [lihat Menyetel format logging biner di Panduan Pengguna Amazon RDS](#).
- Pastikan bahwa log biner tersedia untuk AWS DMS. Karena database yang kompatibel dengan MySQL AWS-managed membersihkan log biner sesegera mungkin, Anda harus menambah lamanya waktu log tetap tersedia. Misalnya, untuk meningkatkan retensi log hingga 24 jam, jalankan perintah berikut.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

- Atur parameter `binlog_format` menjadi "ROW".

Note

Di MySQL atau `binlog_format` MariaDB, adalah parameter dinamis, jadi Anda tidak perlu reboot untuk membuat nilai baru berlaku. Namun, nilai baru hanya akan berlaku untuk sesi baru. Jika Anda beralih `binlog_format` ke ROW untuk tujuan replikasi, database Anda masih dapat membuat log biner berikutnya menggunakan MIXED format, jika sesi tersebut dimulai sebelum Anda mengubah nilainya. Ini dapat AWS DMS mencegah menangkap semua perubahan pada database sumber dengan benar. Saat Anda mengubah `binlog_format` pengaturan pada database MariaDB atau MySQL, pastikan untuk memulai ulang database untuk menutup semua sesi yang ada, atau memulai ulang aplikasi apa pun yang melakukan operasi DHTML (Data Manipulation Language). Memaksa database Anda untuk memulai ulang semua sesi setelah mengubah `binlog_format` parameter ROW akan memastikan bahwa database Anda menulis semua perubahan basis data sumber berikutnya menggunakan format yang benar, sehingga AWS DMS dapat menangkap perubahan tersebut dengan benar.

- Atur parameter `binlog_row_image` ke "Full".
- Atur `binlog_checksum` parameter ke "NONE" untuk DMS versi 3.4.7 atau sebelumnya. Untuk informasi selengkapnya tentang pengaturan parameter di Amazon RDS MySQL, lihat [Menggunakan backup otomatis](#) dalam Panduan Pengguna Amazon RDS.
- Jika Anda menggunakan Amazon RDS MySQL atau Amazon RDS MariaDB baca replika sebagai sumber, aktifkan cadangan pada replika baca, dan pastikan parameter disetel ke `log_slave_updates TRUE`

Batasan dalam menggunakan database MySQL sebagai sumber AWS DMS

Ketika menggunakan basis data MySQL sebagai sumber, pertimbangkan hal berikut:

- Change data capture (CDC) tidak didukung untuk Amazon RDS MySQL 5.5 atau versi yang lebih rendah. Untuk Amazon RDS MySQL, Anda harus menggunakan versi 5.6, 5.7, atau 8.0 untuk mengaktifkan CDC. CDC didukung untuk sumber MySQL 5.5 yang dikelola sendiri.
- Untuk CDC, CREATE TABLE, ADD COLUMN, dan DROP COLUMN mengubah jenis data kolom, dan renaming a columndidukung. Namun, DROP TABLE, RENAME TABLE, dan pembaruan yang dibuat untuk atribut lainnya, seperti nilai default kolom, nullabilitas kolom, set karakter dan sebagainya, tidak didukung.
- Untuk tabel yang dipartisi pada sumber, saat Anda mengatur mode persiapan tabel Target ke Drop tabel pada target, AWS DMS buat tabel sederhana tanpa partisi apa pun pada target MySQL. Untuk memirasi tabel yang dipartisi ke tabel yang dipartisi pada target, buatlah terlebih dahulu tabel yang dipartisi pada basis data MySQL target.
- Menggunakan pernyataan ALTER TABLE *table_name* ADD COLUMN *column_name* untuk menambahkan kolom ke awal (FIRST) atau tengah tabel (AFTER) tidak didukung. Kolom selalu ditambahkan ke akhir tabel.
- CDC tidak didukung ketika nama tabel memuat huruf besar dan huruf kecil, dan mesin sumber di-host pada sistem operasi dengan nama file yang tidak peka kapital. Contohnya adalah Microsoft Windows atau OS X menggunakan HFS+.
- Anda dapat menggunakan Aurora MySQL Edition Serverless v1 yang kompatibel dengan Aurora untuk muatan penuh, tetapi Anda tidak dapat menggunakannya untuk CDC. Ini karena Anda tidak dapat mengaktifkan prasyarat untuk MySQL. Untuk informasi lebih lanjut, lihat [Kelompok parameter dan Aurora Serverless v1](#).

Aurora MySQL yang kompatibel dengan Edition Serverless v2 mendukung CDC.

- Atribut AUTO_INCREMENT pada kolom tidak bermigrasi ke kolom basis data target.
- Menangkap perubahan ketika log biner tidak disimpan di penyimpanan blok standar tidak didukung. Sebagai contoh, CDC tidak bekerja ketika log biner disimpan di Amazon S3.
- AWS DMS membuat tabel target dengan mesin penyimpanan InnoDB secara default. Jika Anda perlu menggunakan mesin penyimpanan selain InnoDB, Anda harus secara manual membuat tabel dan bermigrasi ke mesin tersebut menggunakan mode [do nothing](#).
- Anda tidak dapat menggunakan replika Aurora MySQL sebagai sumber AWS DMS kecuali mode tugas migrasi DMS Anda Migrasikan data yang ada —hanya memuat penuh.

- Jika sumber yang kompatibel dengan MySQL berhenti selama beban penuh, tugas AWS DMS tidak berhenti dengan kesalahan. Tugas berhasil berakhir, tetapi target mungkin tidak sinkron dengan sumber. Jika hal ini terjadi, ulang kembali tugas atau muat ulang tabel yang terpengaruh.
- Indeks yang dibuat pada porsi nilai kolom tidak bermigrasi. Sebagai contoh, indeks `CREATE INDEX first_ten_chars ON pelanggan (name(10))` tidak dibuat pada target.
- Dalam beberapa kasus, tugas dikonfigurasi untuk tidak mereplikasi LOB ("SupportLobs" salah dalam pengaturan tugas atau kolom Jangan sertakan LOB dipilih di konsol tugas). Dalam kasus ini, AWS DMS tidak memigrasikan kolom `MEDIUMBLOB`, `LOB`, `MEDIUMTEXT`, dan `LONGTEXT` apa pun ke target.

Kolom `BLOB`, `TINYBLOB`, `TEXT`, dan `TINYTEXT` tidak terpengaruh dan dimigrasi ke target.

- Tabel data temporal atau tabel sistem-versi tidak didukung pada sumber MariaDB dan basis data target.
- Jika bermigrasi antara dua cluster MySQL Amazon RDS Aurora, titik akhir sumber MySQL RDS Aurora harus berupa instance baca/tulis, bukan instance replika.
- AWS DMS saat ini tidak mendukung migrasi tampilan untuk MariaDB.
- AWS DMS tidak mendukung perubahan DDL untuk tabel yang dipartisi untuk MySQL. Untuk melewati suspensi tabel untuk perubahan DDL partisi selama CDC, atur `skipTableSuspensionForPartitionDdl` ke `true`
- AWS DMS hanya mendukung transaksi XA di versi 3.5.0 dan lebih tinggi. Versi sebelumnya tidak mendukung transaksi XA. AWS DMS tidak mendukung transaksi XA di MariaDB versi 10.6. Untuk informasi lebih lanjut, lihat [the section called "Support untuk transaksi XA"](#) berikut ini.
- AWS DMS tidak menggunakan GTID untuk replikasi, bahkan jika data sumber mengandungnya.
- AWS DMS tidak mendukung kompresi transaksi log biner.
- AWS DMS tidak menyebarkan peristiwa `ON DELETE CASCADE` dan `ON UPDATE CASCADE` untuk database MySQL menggunakan mesin penyimpanan InnoDB. Untuk peristiwa ini, MySQL tidak menghasilkan peristiwa binlog untuk mencerminkan operasi cascaded pada tabel anak. Akibatnya, tidak AWS DMS dapat mereplikasi perubahan yang sesuai ke tabel anak. Untuk informasi selengkapnya, lihat [Indeks, Kunci Asing, atau Pembaruan atau Penghapusan Cascade Tidak Dimigrasi](#).
- AWS DMS tidak menangkap perubahan pada kolom yang dihitung (`VIRTUAL` dan `GENERATED ALWAYS`). Untuk mengatasi batasan ini, lakukan hal berikut:
 - Pra-buat tabel target dalam database target, dan buat AWS DMS tugas dengan `D0_NOthing` atau pengaturan tugas `TRUNCATE_BEFORE_LOAD` beban penuh.

- Tambahkan aturan transformasi untuk menghapus kolom yang dihitung dari cakupan tugas. Untuk informasi tentang aturan transformasi, lihat [Aturan dan tindakan transformasi](#).

Support untuk transaksi XA

Transaksi Extended Architecture (XA) adalah transaksi yang dapat digunakan untuk mengelompokkan serangkaian operasi dari beberapa sumber daya transaksional menjadi satu transaksi global yang andal. Transaksi XA menggunakan protokol komit dua fase. Secara umum, menangkap perubahan saat ada transaksi XA terbuka dapat menyebabkan hilangnya data. Jika database Anda tidak menggunakan transaksi XA, Anda dapat mengabaikan izin ini dan konfigurasi `IgnoreOpenXaTransactionsCheck` dengan menggunakan nilai `true`. Untuk mulai mereplikasi dari sumber yang memiliki transaksi XA, lakukan hal berikut:

- Pastikan bahwa pengguna AWS DMS endpoint memiliki izin berikut:

```
grant XA_RECOVER_ADMIN on *.* to 'userName'@'%';
```

- Atur pengaturan titik akhir `IgnoreOpenXaTransactionsCheck` ke `false`.

Note

AWS DMS tidak mendukung transaksi XA pada MariaDB Source DB versi 10.6.

Pengaturan titik akhir saat menggunakan MySQL sebagai sumber AWS DMS

Anda dapat menggunakan pengaturan endpoint untuk mengkonfigurasi database sumber MySQL Anda mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir sumber menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--my-sql-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan endpoint yang dapat Anda gunakan dengan MySQL sebagai sumber.

Nama	Deskripsi
EventsPollInterval	<p>Menentukan seberapa sering memeriksa log biner untuk perubahan/peristiwa baru ketika basis data diam.</p> <p>Nilai default: 5</p> <p>Nilai yang valid: 1–60.</p> <p>Contoh: <code>--my-sql-settings '{"EventsPollInterval": 5}'</code></p> <p>Dalam contoh, AWS DMS memeriksa perubahan dalam log biner setiap lima detik.</p>
ExecuteTimeout	<p>Untuk AWS DMS versi 3.4.7 dan yang lebih tinggi, menetapkan batas waktu pernyataan klien untuk titik akhir sumber MySQL, dalam hitungan detik.</p> <p>Nilai default: 60</p> <p>Contoh: <code>--my-sql-settings '{"ExecuteTimeout": 1500}'</code></p>
ServerTimezone	<p>Menentukan zona waktu untuk basis data sumber MySQL.</p> <p>Contoh: <code>--my-sql-settings '{"ServerTimezone": "US/Pacific"}</code></p>
AfterConnectScript	<p>Menentukan script untuk menjalankan segera setelah AWS DMS menghubungkan ke endpoint. Tugas migrasi terus berjalan terlepas jika pernyataan SQL berhasil atau gagal.</p> <p>Nilai yang valid: Satu atau lebih pernyataan SQL yang valid, yang dimulai dengan titik koma.</p>

Nama	Deskripsi
<p>CleanSrcMetadataOnMismatch</p>	<p>Contoh: <code>--my-sql-settings '{"AfterConnectScript": "ALTER SESSION SET CURRENT_SCHEMA=system"}'</code></p> <p>Membersihkan dan membuat kembali tabel metadata informasi pada instans replikasi ketika terjadi ketidakcocokan. Misalnya, dalam situasi di mana menjalankan alter DDL di tabel dapat mengakibatkan informasi yang berbeda mengenai tabel cache dalam instans replikasi.</p> <p>. Boolean</p> <p>Nilai default: <code>false</code></p> <p>Contoh: <code>--my-sql-settings '{"CleanSrcMetadataOnMismatch": false}'</code></p>
<p>skipTableSuspensionForPartitionDdl</p>	<p>AWS DMS tidak mendukung perubahan DDL untuk tabel yang dipartisi untuk MySQL. Untuk AWS DMS versi 3.4.6 dan yang lebih tinggi, atur ini untuk <code>true</code> melewati suspensi tabel untuk perubahan DDL partisi selama CDC. AWS DMS mengabaikan <code>partitioned-table-related</code> DDL, dan terus memproses perubahan log biner lebih lanjut.</p> <p>Nilai default: <code>false</code></p> <p>Contoh: <code>--my-sql-settings '{"skipTableSuspensionForPartitionDdl": true}'</code></p>
<p>IgnoreOpenXaTransactionsCheck</p>	<p>Untuk AWS DMS versi 3.5.0 dan yang lebih tinggi, tentukan apakah tugas harus mengabaikan transaksi XA terbuka saat memulai. Setel ini ke <code>false</code> jika sumber Anda memiliki transaksi XA.</p> <p>Nilai default: <code>true</code></p> <p>Contoh: <code>--my-sql-settings '{"IgnoreOpenXaTransactionsCheck": false}'</code></p>

Jenis data sumber untuk MySQL

Tabel berikut menunjukkan tipe data sumber database MySQL yang didukung saat AWS DMS menggunakan dan pemetaan AWS DMS default dari tipe data.

Untuk informasi tentang cara melihat jenis data yang dipetakan dalam target, lihat bagian titik akhir target yang Anda gunakan.

Untuk informasi tambahan tentang tipe AWS DMS data, lihat [Tipe data untuk AWS Database Migration Service](#).

Tipe data MySQL	AWS DMS tipe data
INT	INT4
BIGINT	INT8
MEDIUMINT	INT4
TINYINT	INT1
SMALLINT	INT2
UNSIGNED TINYINT	UINT1
UNSIGNED SMALLINT	UINT2
MEDIUMINT YANG TIDAK DITANDATANGANI	UINT4
INT TIDAK DITANDATANGANI	UINT4
UNSIGNED BIGINT	UINT8
DECIMAL(10)	NUMERIC (10,0)
BINARY	BYTES(1)
BIT	BOOLEAN
BIT(64)	BYTES(8)
BLOB	BYTE (65535)

Tipe data MySQL	AWS DMS tipe data
LONGBLOB	BLOB
MEDIUMBLOB	BLOB
TINYBLOB	BYTES(255)
DATE	DATE
DATETIME	DATETIME DATETIME tanpa nilai kurung direplikasi tanpa milidetik. DATETIME dengan nilai kurung 1 sampai 5 (seperti DATETIME(5)) direplikasi dengan milidetik. Saat mereplikasi kolom DATETIME, waktunya tetap sama pada target. Itu tidak dikonversi ke UTC.
TIME	STRING
TIMESTAMP	DATETIME Saat mereplikasi kolom TIMESTAMP, waktu dikonversi ke UTC pada target.
YEAR	INT2
DOUBLE	REAL8

Tipe data MySQL	AWS DMS tipe data
FLOAT	REAL(DOUBLE) Jika nilai FLOAT tidak berada dalam kisaran berikut, gunakan transformasi untuk memetakan FLOAT ke STRING. Untuk informasi lebih lanjut tentang transformasi, lihat Aturan dan tindakan transformasi . Rentang FLOAT yang didukung adalah -1.79E+308 hingga -2.23E-308, 0, dan 2.23E-308 hingga 1.79E+308
VARCHAR (45)	WSTRING (45)
VARCHAR (2000)	WSTRING (2000)
VARCHAR (4000)	WSTRING (4000)
VARBINARY (4000)	BYTES (4000)
VARBINARY (2000)	BYTE (2000)
CHAR	WSTRING
TEXT	WSTRING
LONGTEXT	NCLOB
MEDIUMTEXT	NCLOB
TINYTEXT	WSTRING (255)
GEOMETRY	BLOB
POINT	BLOB
LINestring	BLOB
POLYGON	BLOB

Tipe data MySQL	AWS DMS tipe data
MULTIPOINT	BLOB
MULTILINESTRING	BLOB
MULTIPOLYGON	BLOB
GEOMETRYCOLLECTION	BLOB
ENUM	WSTRING (<i>length</i>) Di sini, <i>length</i> adalah panjang dari nilai terpanjang di ENUM.
SET	WSTRING (<i>length</i>) Di sini, <i>length</i> adalah total panjang semua nilai dalam SET, termasuk koma.
JSON	CLOB

Note

Dalam beberapa kasus, Anda mungkin menentukan jenis data DATETIME dan TIMESTAMP dengan nilai “no” (yaitu 0000-00-00). Jika demikian, pastikan bahwa basis data target dalam tugas replikasi mendukung nilai-nilai “no” untuk jenis data DATETIME dan TIMESTAMP. Jika tidak, nilai-nilai ini dicatat sebagai null pada target.

Menggunakan basis data SAP ASE sebagai sumber untuk AWS DMS

Anda dapat memigrasikan data dari basis data SAP Adaptive Server Enterprise (ASE) — yang sebelumnya dikenal sebagai SYBAS—menggunakan AWS DMS. Dengan database SAP ASE sebagai sumber, Anda dapat memigrasi data ke salah satu sumber lain yang didukung AWS DMS basis data target.

Untuk informasi tentang versi SAP ASE yang AWS DMS mendukung sebagai sumber, lihat [Sumber untuk AWS DMS](#).

Untuk detail tambahan tentang menggunakan basis data SAP ASE dan AWS DMS, lihat bagian berikut.

Topik

- [Prasyarat untuk menggunakan basis data SAP ASE sebagai sumber untuk AWS DMS](#)
- [Keterbatasan penggunaan SAP ASE sebagai sumber untuk AWS DMS](#)
- [Izin yang diperlukan untuk menggunakan SAP ASE sebagai sumber untuk AWS DMS](#)
- [Menghapus titik pemotongan](#)
- [Pengaturan titik akhir ketika menggunakan SAP ASE sebagai sumber untuk AWS DMS](#)
- [Jenis data sumber untuk SAP ASE](#)

Prasyarat untuk menggunakan basis data SAP ASE sebagai sumber untuk AWS DMS

Untuk basis data SAP ASE sebagai sumber untuk AWS DMS, lakukan hal berikut:

- Aktifkan replikasi SAP ASE untuk tabel dengan menggunakan perintah `sp_setreptable`. Untuk informasi selengkapnya, lihat [Sybase Infocenter Archive](#).
- Nonaktifkan RepAgent pada basis data SAP ASE. Untuk informasi selengkapnya, lihat [Hentikan dan nonaktifkan RepAgent thread dalam database utama](#).
- Untuk mereplikasi ke SAP ASE versi 15.7 pada Windows instans EC2 yang dikonfigurasi untuk karakter non-Latin (misalnya, aksara Cina), instal SAP ASE 15.7 SP121 pada komputer target.

Note

Untuk replikasi change data capture (CDC) berkelanjutan, DMS menjalankan `dbcc logtransfer` dan `dbcc logunlock` untuk membaca data dari log transaksi.

Keterbatasan penggunaan SAP ASE sebagai sumber untuk AWS DMS

Keterbatasan berikut berlaku saat menggunakan basis data SAP ASE sebagai sumber untuk AWS DMS:

- Anda dapat menjalankan hanya satu AWS DMS tugas dengan replikasi berkelanjutan atau CDC untuk setiap database SAP ASE. Anda dapat menjalankan beberapa full-load-only tugas secara paralel.

Izin yang diperlukan untuk menggunakan SAP ASE sebagai sumber untuk AWS DMS

Untuk menggunakan basis data SAP ASE sebagai sumber dalam tugas AWS DMS, Anda perlu memberikan izin. Berikan akun pengguna yang ditentukan dalam definisi izin basis data AWS DMS berikut dalam basis data SAP ASE:

- sa_role
- replication_role
- sybase_ts_role
- Secara default, di mana Anda harus memiliki izin untuk menjalankan prosedur `sp_setreptable` yang tersimpan, AWS DMS memungkinkan opsi replikasi SAP ASE. Jika Anda ingin menjalankan `sp_setreptable` pada tabel langsung dari titik akhir basis data dan tidak melalui AWS DMS, Anda dapat menggunakan atribut sambungan tambahan `enableReplication`. Untuk informasi selengkapnya, lihat [Pengaturan titik akhir ketika menggunakan SAP ASE sebagai sumber untuk AWS DMS](#).

Menghapus titik pemotongan

Ketika tugas dimulai, AWS DMS menetapkan entri `$replication_truncation_point` di tampilan sistem `syslogshold`, menunjukkan bahwa proses replikasi sedang berlangsung. Sementara AWS DMS bekerja, DMS memajukan titik pemotongan replikasi secara berkala, sesuai dengan jumlah data yang telah disalin ke target.

Setelah entri `$replication_truncation_point` didirikan, jangan biarkan tugas AWS DMS berjalan untuk mencegah log basis data menjadi terlalu besar. Jika Anda ingin menghentikan tugas AWS DMS secara permanen, hapus titik pemotongan replikasi dengan mengeluarkan perintah berikut:

```
dbcc settrunc('lrm','ignore')
```

Setelah titik pemotongan dihapus, Anda tidak dapat melanjutkan tugas AWS DMS. Log terus dipotong secara otomatis di pos pemeriksaan (jika pemotongan otomatis diatur).

Pengaturan titik akhir ketika menggunakan SAP ASE sebagai sumber untuk AWS DMS

Anda dapat menggunakan pengaturan titik akhir untuk mengonfigurasi database sumber SAP ASE serupa dengan ketika menggunakan atribut koneksi tambahan. Anda menentukan pengaturan ketika membuat titik akhir sumber ketika menggunakan titik akhir sumber ketika menggunakan titik akhir

sumber AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan `--sybase-settings '{"EndpointSetting": "value", ...}'` Sintaks JSON.

Tabel berikut menunjukkan pengaturan titik akhir yang dapat Anda gunakan dengan SAP ASE sebagai sumber.

Nama	Deskripsi
Charset	<p>Tetapkan atribut ini ke nama SAP ASE yang sesuai dengan rangkaian karakter internasional.</p> <p>Nilai default: <code>iso_1</code></p> <p>Contoh: <code>--sybase-settings '{"Charset": "utf8"}'</code></p> <p>Nilai yang valid:</p> <ul style="list-style-type: none"> • <code>acsii_8</code> • <code>big5hk</code> • <code>cp437</code> • <code>cp850</code> • <code>cp852</code> • <code>cp852</code> • <code>cp855</code> • <code>cp857</code> • <code>cp858</code> • <code>cp860</code> • <code>cp864</code> • <code>cp866</code> • <code>cp869</code> • <code>cp874</code> • <code>cp932</code> • <code>cp936</code> • <code>cp950</code>

Nama	Deskripsi
	<ul style="list-style-type: none">• cp1250• cp1251• cp1252• cp1253• cp1254• cp1255• cp1256• cp1257• cp1258• deckanji• euccns• eucgb• eucjis• eucksc• gb18030• greek8• iso_1• iso88592• iso88595• iso88596• iso88597• iso88598• iso88599• iso15• kz1048• koi8• roman8• iso88599• sjis

Nama	Deskripsi
	<ul style="list-style-type: none"> • tis620 • turkish8 • utf8 <p>Untuk pertanyaan lebih lanjut tentang rangkaian karakter yang didukung dalam basis data SAP ASE, lihat Adaptive Server Enterprise: Rangkaian Karakter yang Didukung.</p>
EnableReplication	<p>Atur atribut ini jika Anda ingin mengaktifkan <code>sp_setrep table</code> pada tabel dari akhir basis data dan tidak melalui AWS DMS.</p> <p>Nilai default: <code>true</code></p> <p>Nilai yang valid: <code>true</code> atau <code>false</code></p> <p>Contoh: <code>--sybase-settings '{"Enable Replication": false}'</code></p>
EncryptPassword	<p>Atur atribut ini jika Anda telah mengaktifkan <code>"net password encryption reqd"</code> pada sumber basis data.</p> <p>Nilai default: <code>0</code></p> <p>Nilai valid: <code>0</code>, <code>1</code>, atau <code>2</code></p> <p>Contoh: <code>--sybase-settings '{"EncryptPassword": 1}'</code></p> <p>Untuk informasi lebih lanjut tentang nilai parameter ini, lihat Adaptive Server Enterprise: Menggunakan EncryptPassword Properti string koneksi.</p>

Nama	Deskripsi
Provider	<p>Anda dapat menggunakan Transport Layer Security (TLS) 1.2 untuk versi ASE 15.7 dan lebih tinggi. Perhatikan bahwa AWS membutuhkan TLS versi 1.2 atau yang lebih baru, dan merekomendasikan versi 1.3.</p> <p>Nilai default: Adaptive Server Enterprise</p> <p>Nilai valid: Adaptive Server Enterprise 16.03.06</p> <p>Contoh: <code>--sybase-settings '{"Provider": "Adaptive Server Enterprise 16.03.06"}'</code></p>

Jenis data sumber untuk SAP ASE

Untuk daftar jenis data sumber SAP ASE yang didukung saat menggunakan AWS DMS dan pemetaan default dari jenis data AWS DMS, lihat tabel berikut. AWS DMS tidak mendukung tabel sumber SAP ASE dengan kolom jenis data user-defined type (UDT). Kolom yang direplikasi dengan jenis data ini dibuat sebagai NULL.

Untuk informasi tentang cara untuk melihat jenis data yang dipetakan dalam target, lihat bagian [Target migrasi data](#) untuk titik akhir target Anda.

Untuk informasi tambahan tentang jenis data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Jenis data SAP ASE	Jenis data AWS DMS
BIGINT	INT8
UNSIGNED BIGINT	UINT8
INT	INT4
INT TIDAK DITANDATANGANI	UINT4
SMALLINT	INT2

Jenis data SAP ASE	Jenis data AWS DMS
UNSIGNED SMALLINT	UINT2
TINYINT	UINT1
DESIMAL	NUMERIK
NUMERIK	NUMERIK
MENGAMBANG	REAL8
DOUBLE	REAL8
REAL	REAL4
MONEY	NUMERIK
SMALLMONEY	NUMERIK
DATETIME	DATETIME
BIGDATETIME	TANGGAL WAKTU (6)
SMALLDATETIME	DATETIME
DATE	DATE
TIME	WAKTU
BIGTIME	WAKTU
CHAR	STRING
UNICHAR	WSTRING
NCHAR	WSTRING
VARCHAR	STRING
UNIVARCHAR	WSTRING

Jenis data SAP ASE	Jenis data AWS DMS
NVARCHAR	WSTRING
BINER	BYTES
VARBINARY	BYTES
BIT	BOOLEAN
TEXT	CLOB
UNITEXT	NCLOB
IMAGE	BLOB

Menggunakan MongoDB sebagai sumber untuk AWS DMS

Untuk informasi tentang versi MongoDB AWS DMS yang mendukung sebagai sumber, lihat. [Sumber untuk AWS DMS](#)

Perhatikan hal berikut tentang dukungan versi MongoDB:

- Versi AWS DMS 3.4.5 dan yang lebih baru mendukung MongoDB versi 4.2 dan 4.4.
- Versi AWS DMS 3.4.5 dan yang lebih baru dan versi MongoDB 4.2 dan yang lebih baru mendukung transaksi terdistribusi. [Untuk informasi selengkapnya tentang transaksi terdistribusi MongoDB, lihat Transaksi dalam dokumentasi MongoDB.](#)
- Versi AWS DMS 3.5.0 dan yang lebih baru tidak mendukung versi MongoDB sebelum 3.6.
- Versi AWS DMS 3.5.1 dan yang lebih baru mendukung MongoDB versi 5.0.
- Versi AWS DMS 3.5.2 dan yang lebih baru mendukung MongoDB versi 6.0.

Jika Anda baru menggunakan MongoDB, ketahuilah konsep basis data MongoDB penting berikut:

- Catatan di MongoDB adalah dokumen, yang merupakan struktur data terdiri dari pasangan bidang dan nilai. Nilai bidang dapat mencakup dokumen, rangkaian, dan rangkaian dokumen lain. Dokumen kurang lebih setara dengan baris dalam tabel basis data relasional.

- Koleksi di MongoDB adalah sekelompok dokumen, dan kurang lebih setara dengan tabel basis data relasional.
- Database di MongoDB adalah sekumpulan koleksi, dan kira-kira setara dengan skema dalam database relasional.
- Secara internal, dokumen MongoDB disimpan sebagai file biner JSON (BSON) dalam format terkompresi yang mencakup jenis untuk setiap bidang dalam dokumen. Setiap dokumen mempunyai ID yang unik.

AWS DMS mendukung dua mode migrasi saat menggunakan MongoDB sebagai sumber, Mode dokumen atau Mode tabel. Anda menentukan mode migrasi mana yang digunakan ketika Anda membuat titik akhir MongoDB atau dengan menetapkan parameter Mode metadata dari konsol AWS DMS. Secara fakultatif, Anda dapat membuat kolom kedua bernama `_id` yang bertindak sebagai kunci utama dengan memilih tombol tanda centang untuk `_id` sebagai kolom terpisah di panel konfigurasi titik akhir.

Pilihan mode migrasi Anda memengaruhi format yang dihasilkan dari data target, seperti yang dijelaskan berikut.

Mode dokumen

Dalam mode dokumen, dokumen MongoDB bermigrasi dengan apa adanya, artinya bahwa data dokumen dikonsolidasikan ke dalam satu kolom bernama `_doc` dalam tabel target. Mode dokumen adalah pengaturan default ketika Anda menggunakan MongoDB sebagai titik akhir sumber.

Misalnya, pertimbangkan dokumen-dokumen berikut dalam koleksi MongoDB yang disebut `MyCollection`.

```
> db.myCollection.find()
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe0"), "a" : 1, "b" : 2, "c" : 3 }
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe1"), "a" : 4, "b" : 5, "c" : 6 }
```

Setelah memigrasikan data ke tabel basis data relasional menggunakan mode dokumen, data terstruktur sebagai berikut. Bidang data dalam dokumen MongoDB dikonsolidasikan ke dalam kolom `_doc`.

oid_id	_doc
--------	------

5a94815f40bd44d1b02bdfe0	{ "a" : 1, "b" : 2, "c" : 3 }
5a94815f40bd44d1b02bdfe1	{ "a" : 4, "b" : 5, "c" : 6 }

Anda dapat mengatur atribut sambungan tambahan secara fakultatif `extractDocID` menjadi benar untuk membuat kolom kedua bernama `"_id"` yang bertindak sebagai kunci utama. Jika Anda akan menggunakan CDC, atur parameter ini ke `true`.

Dalam mode dokumen, AWS DMS mengelola pembuatan dan penggantian nama koleksi seperti ini:

- Jika Anda menambahkan koleksi baru ke basis data sumber, AWS DMS menciptakan tabel target baru untuk koleksi dan mereplikasi dokumen.
- Jika Anda mengubah nama koleksi yang ada pada basis data sumber, AWS DMS tidak mengubah nama tabel target.

Jika titik akhir target adalah Amazon DocumentDB, jalankan migrasi dalam Mode dokumen.

Mode tabel

Dalam mode tabel, AWS DMS mengubah setiap bidang tingkat atas dalam dokumen MongoDB menjadi kolom dalam tabel target. Jika bidang bersarang, AWS DMS meratakan nilai-nilai bersarang ke dalam satu kolom. AWS DMS kemudian menambahkan bidang kunci dan jenis data untuk tabel target rangkaian kolom.

Untuk setiap dokumen MongoDB, AWS DMS menambahkan setiap kunci dan jenis pada tabel target rangkaian kolom. Misalnya, dengan menggunakan mode tabel, AWS DMS memigrasikan contoh sebelumnya ke dalam tabel berikut.

oid_id	a	b	c
5a94815f40bd44d1b02bdfe0	1	2	3
5a94815f40bd44d1b02bdfe1	4	5	6

Nilai-nilai bersarang diratakan menjadi sebuah kolom yang berisi nama-nama kunci yang dipisahkan oleh titik. Kolom dinamakan penggabungan nama bidang yang diratakan yang

dipisahkan oleh titik. Misalnya, AWS DMS memigrasi dokumen JSON dengan bidang nilai-nilai bersarang seperti `{"a" : {"b" : {"c": 1}}}` ke dalam sebuah kolom bernama `a.b.c`.

Untuk membuat kolom target, AWS DMS memindai sejumlah dokumen MongoDB tertentu dan menciptakan rangkaian semua bidang dan jenis mereka. AWS DMS kemudian menggunakan rangkaian ini untuk membuat kolom dari tabel target. Jika Anda membuat atau memodifikasi titik akhir sumber MongoDB Anda menggunakan konsol, Anda dapat menentukan jumlah dokumen yang dipindai. Nilai default adalah 1000 dokumen. Jika Anda menggunakan AWS CLI, Anda dapat menggunakan atribut sambungan tambahan `docsToInvestigate`.

Dalam mode tabel, AWS DMS mengelola dokumen dan koleksi seperti ini:

- Ketika Anda menambahkan dokumen ke koleksi yang ada, dokumen itu direplikasi. Jika ada bidang yang tidak ada dalam target, bidang tersebut tidak direplikasi.
- Ketika Anda memperbarui dokumen, dokumen yang diperbarui tersebut direplikasi. Jika ada bidang yang tidak ada dalam target, bidang tersebut tidak direplikasi.
- Menghapus dokumen sepenuhnya didukung.
- Menambahkan koleksi baru tidak menghasilkan tabel baru pada target ketika dilakukan selama tugas CDC.
- Pada fase Change Data Capture (CDC), AWS DMS tidak mendukung penggantian nama koleksi.

Topik

- [Izin yang dibutuhkan saat menggunakan MongoDB sebagai sumber untuk AWS DMS](#)
- [Mengonfigurasi rangkaian replika MongoDB yang untuk CDC](#)
- [Persyaratan keamanan saat menggunakan MongoDB sebagai sumber untuk AWS DMS](#)
- [Melakukan segmen koleksi MongoDB dan bermigrasi secara paralel](#)
- [Migrasi beberapa database saat menggunakan MongoDB sebagai sumber AWS DMS](#)
- [Keterbatasan saat menggunakan MongoDB sebagai sumber untuk AWS DMS](#)
- [Pengaturan konfigurasi titik akhir saat menggunakan MongoDB sebagai sumber untuk AWS DMS](#)
- [Jenis data sumber untuk MongoDB](#)

Izin yang dibutuhkan saat menggunakan MongoDB sebagai sumber untuk AWS DMS

Untuk migrasi AWS DMS dengan sumber MongoDB, Anda dapat membuat baik akun pengguna dengan hak istimewa root, atau pengguna dengan izin hanya pada basis data untuk bermigrasi.

Kode berikut membuat pengguna menjadi akun root.

```
use admin
db.createUser(
  {
    user: "root",
    pwd: "password",
    roles: [ { role: "root", db: "admin" } ]
  }
)
```

Untuk sumber MongoDB 3.x, kode berikut menciptakan pengguna dengan hak izin minimal pada basis data yang akan bermigrasi.

```
use database_to_migrate
db.createUser(
  {
    user: "dms-user",
    pwd: "password",
    roles: [ { role: "read", db: "local" }, "read" ]
  })
```

Untuk sumber 4.x MongoDB, kode berikut membuat pengguna dengan hak izin minimal.

```
{ resource: { db: "", collection: "" }, actions: [ "find", "changeStream" ] }
```

Sebagai contoh, buatlah peran berikut dalam basis data "admin".

```
use admin
db.createRole(
  {
    role: "changestreamrole",
    privileges: [
      { resource: { db: "", collection: "" }, actions: [ "find","changeStream" ] }
    ],
    roles: []
  })
```

```
}  
)
```

Dan setelah peran itu dibuat, buatlah pengguna dalam basis data yang akan bermigrasi.

```
> use test  
> db.createUser(  
  {  
    user: "dms-user12345",  
    pwd: "password",  
    roles: [ { role: "changestreamrole", db: "admin" }, "read"]  
  })
```

Mengonfigurasi rangkaian replika MongoDB yang untuk CDC

Untuk menggunakan replikasi yang sedang berlangsung atau CDC dengan MongoDB, AWS DMS membutuhkan akses ke operations log (oplog) MongoDB. Untuk membuat oplog, Anda perlu men-deploy rangkaian replika jika salah satu tidak ada. Untuk informasi selengkapnya tentang penggunaan, lihat [dokumentasi MongoDB](#).

Anda dapat menggunakan CDC dengan node primer atau node sekunder dari rangkaian replika MongoDB sebagai titik akhir sumber.

Mengonversi instans mandiri menjadi rangkaian replika

1. Menggunakan baris perintah, hubungkan ke mongo.

```
mongo localhost
```

2. Hentikan layanan mongod.

```
service mongod stop
```

3. Mulai ulang mongod menggunakan perintah berikut ini:

```
mongod --replSet "rs0" --auth -port port_number
```

4. Uji sambungan ke rangkaian replika menggunakan perintah berikut:

```
mongo -u root -p password --host rs0/localhost:port_number  
  --authenticationDatabase "admin"
```

Jika Anda berencana untuk melakukan migrasi mode dokumen, pilih opsi `_id as a separate column` ketika Anda membuat titik akhir MongoDB. Memilih opsi ini membuat kolom kedua bernama `_id` yang bertindak sebagai kunci utama. Kolom kedua ini diperlukan oleh AWS DMS untuk mendukung operasi data manipulation language (DML).

Note

AWS DMS menggunakan log operasi (oplog) untuk menangkap perubahan selama replikasi yang sedang berlangsung. Jika MongoDB mengeluarkan catatan dari oplog AWS DMS sebelum membacanya, tugas Anda gagal. Kami merekomendasikan ukuran oplog untuk mempertahankan perubahan setidaknya selama 24 jam.

Persyaratan keamanan saat menggunakan MongoDB sebagai sumber untuk AWS DMS

AWS DMS mendukung dua metode autentikasi untuk MongoDB. Dua metode autentikasi digunakan untuk mengenkripsi kata sandi, sehingga mereka hanya digunakan ketika parameter `authType` diatur menjadi `PASSWORD`.

Metode autentikasi MongoDB adalah sebagai berikut:

- `MONGODB-CR` - Untuk kompatibilitas mundur
- `SCRAM-SHA-1` - Default saat menggunakan MongoDB versi 3.x dan 4.0

Jika metode autentikasi tidak ditentukan, AWS DMS menggunakan metode default untuk versi sumber MongoDB.

Melakukan segmen koleksi MongoDB dan bermigrasi secara paralel

Untuk meningkatkan kinerja tugas migrasi, titik akhir sumber MongoDB mendukung dua opsi untuk beban penuh paralel dalam pemetaan tabel.

Dengan kata lain, Anda dapat memigrasikan koleksi secara paralel dengan menggunakan autosegmentasi atau segmentasi rentang dengan pemetaan tabel untuk beban penuh paralel dalam pengaturan JSON. Dengan autosegmentasi, Anda dapat menentukan kriteria AWS DMS untuk secara otomatis mengelompokkan sumber Anda untuk migrasi di setiap utas. Dengan segmentasi rentang, Anda dapat memberi tahu AWS DMS rentang spesifik setiap segmen untuk DMS untuk

bermigrasi di setiap utas. Untuk informasi selengkapnya tentang pengaturan, lihat [Tabel dan koleksi pengaturan aturan dan operasi](#).

Migrasi database MongoDB secara paralel menggunakan rentang autosegmentasi

Anda dapat memigrasikan dokumen secara paralel dengan menentukan kriteria untuk secara otomatis AWS DMS mempartisi (segmen) data Anda untuk setiap utas. Secara khusus, Anda menentukan jumlah dokumen yang akan dimigrasikan per utas. Dengan menggunakan pendekatan ini, AWS DMS upaya untuk mengoptimalkan batas segmen untuk kinerja maksimum per utas.

Anda dapat menentukan kriteria segmentasi menggunakan opsi pengaturan tabel berikut dalam pemetaan tabel.

Opsi tabel-pengaturan	Deskripsi
"type"	(Wajib) Setel ke "partitions-auto" untuk MongoDB sebagai sumber.
"number-of-partitions"	(Opsional) Jumlah total partisi (segmen) yang digunakan untuk migrasi. Defaultnya adalah 16.
"collection-count-from-metadata"	(Opsional) Jika opsi ini diatur ke <code>true</code> , AWS DMS menggunakan perkiraan jumlah koleksi untuk menentukan jumlah partisi. Jika opsi ini disetel ke <code>false</code> , AWS DMS gunakan jumlah koleksi yang sebenarnya. Default-nya adalah <code>true</code> .
"max-records-skip-per-page"	(Opsional) Jumlah catatan yang harus dilewati sekaligus saat menentukan batas untuk setiap partisi. AWS DMS menggunakan pendekatan <code>skip paginated</code> untuk menentukan batas minimum untuk partisi. Default adalah 10.000. Menyetel nilai yang relatif besar dapat mengakibatkan batas waktu kursor dan kegagalan tugas. Menetapkan nilai yang relatif rendah menghasilkan lebih banyak operasi per halaman dan beban penuh yang lebih lambat.

Opsi tabel-pengaturan	Deskripsi
"batch-size"	(Opsional) Membatasi jumlah dokumen yang dikembalikan dalam satu batch. Setiap batch membutuhkan perjalanan pulang pergi ke server. Jika ukuran batch nol (0), kursor menggunakan ukuran batch maksimum yang ditentukan server. Default-nya adalah 0.

Contoh berikut menunjukkan pemetaan tabel untuk autosegmentasi.

```
{
  "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
      }
    }
  ]
}
```

```
}
```

Autosegmentasi memiliki batasan berikut. Migrasi untuk setiap segmen mengambil jumlah koleksi dan minimum `_id` untuk koleksi secara terpisah. Kemudian menggunakan lompatan paginasi untuk menghitung batas minimum untuk segmen itu.

Oleh karena itu, pastikan bahwa `_id` nilai minimum untuk setiap koleksi tetap konstan sampai semua batas segmen dalam koleksi dihitung. Jika Anda mengubah `_id` nilai minimum untuk koleksi selama perhitungan batas segmennya, itu dapat menyebabkan kehilangan data atau kesalahan baris duplikat.

Migrasi database MongoDB secara paralel menggunakan segmentasi rentang

Anda dapat memigrasikan dokumen secara paralel dengan menentukan rentang untuk setiap segmen dalam utas. Dengan menggunakan pendekatan ini, Anda memberi tahu AWS DMS dokumen tertentu untuk bermigrasi di setiap utas sesuai dengan pilihan rentang dokumen per utas.

Gambar berikut menunjukkan koleksi MongoDB yang memiliki tujuh item, `_id` dan sebagai kunci utama.

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

Untuk membagi koleksi menjadi tiga segmen tertentu AWS DMS untuk bermigrasi secara paralel, Anda dapat menambahkan aturan pemetaan tabel ke tugas migrasi Anda. Pendekatan ini ditunjukkan dalam contoh JSON berikut.

```
{ // 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

```

Bahwa definisi pemetaan tabel membagi koleksi sumber menjadi tiga segmen dan bermigrasi secara paralel. Berikut ini adalah batas-batas segmentasi.

```
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)
```

Setelah tugas migrasi selesai, Anda dapat melakukan verifikasi dari log tugas yang dimuat tabel secara paralel, seperti yang ditunjukkan dalam contoh berikut. Anda juga dapat memverifikasi klausa `find` MongoDB yang digunakan untuk membongkar setiap segmen dari tabel sumber.

```
[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
transferred 480.
```

```
[TASK_MANAGER ] I: Load finished for segment #1 of table  
'testdatabase'. 'testtable' (Id = 1) by subtask 1. 2 records transferred.
```

Saat ini, AWS DMS mendukung tipe data MongoDB berikut sebagai kolom kunci segmen:

- Dua Kali
- String
- ObjectId
- 32 bit integer
- 64 bit integer

Migrasi beberapa database saat menggunakan MongoDB sebagai sumber AWS DMS

AWS DMS versi 3.4.5 dan yang lebih tinggi mendukung migrasi beberapa database dalam satu tugas untuk semua versi MongoDB yang didukung. Jika Anda ingin memigrasi beberapa database, lakukan langkah-langkah berikut:

1. Saat Anda membuat titik akhir sumber MongoDB, lakukan salah satu hal berikut:
 - Pada halaman Create endpoint konsol DMS, pastikan nama Database kosong di bawah konfigurasi Endpoint.
 - Dengan menggunakan AWS CLI `CreateEndpoint` perintah, tetapkan nilai string kosong ke `DatabaseName` parameter di `MongoDBSettings`.
2. Untuk setiap database yang ingin Anda migrasi dari sumber MongoDB, tentukan nama database sebagai nama skema dalam pemetaan tabel untuk tugas tersebut. Anda dapat melakukan ini menggunakan input yang dipandu di konsol atau langsung di JSON. Untuk informasi lebih lanjut tentang input yang dipandu, lihat [Menentukan pemilihan tabel dan transformasi aturan dari konsol](#). Untuk informasi lebih lanjut tentang JSON, lihat [Aturan seleksi dan tindakan](#).

Misalnya, Anda dapat menentukan JSON berikut untuk memigrasikan tiga database MongoDB.

Example Memigrasi semua tabel dalam skema

JSON berikut memigrasikan semua tabel dari `Customers`, `Orders`, dan `Suppliers` database di titik akhir sumber Anda ke titik akhir target Anda.

```
{
  "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": []
    }
  ]
}
```

Keterbatasan saat menggunakan MongoDB sebagai sumber untuk AWS DMS

Berikut ini adalah keterbatasan ketika menggunakan MongoDB sebagai sumber untuk AWS DMS:

- Dalam mode tabel, dokumen dalam koleksi harus konsisten dalam tipe data yang mereka gunakan untuk nilai di bidang yang sama. Misalnya, jika dokumen dalam koleksi termasuk '{ a : { b: value ... } }', semua dokumen dalam koleksi yang mereferensikan *value* a.b bidang harus menggunakan tipe data yang sama untuk *value*, di mana pun itu muncul dalam koleksi.
- Saat opsi `_id` diatur sebagai kolom terpisah, string ID tidak dapat melebihi 200 karakter.
- ID objek dan kunci jenis rangkaian dikonversi ke kolom yang diawali dengan `oid` dan `array` dalam mode tabel.

Secara internal, kolom ini diacu dengan nama prefiks. Jika Anda menggunakan aturan transformasi di AWS DMS yang mengacu kolom ini, pastikan untuk menentukan kolom prefiks. Misalnya, Anda menentukan `$_oid__id` dan bukan `$_id`, atau `$_array__addresses` dan bukan `$_addresses`.

- Nama koleksi dan nama kunci tidak dapat menyertakan simbol dolar (\$).
- AWS DMS tidak mendukung koleksi yang berisi bidang yang sama dengan kasus yang berbeda (atas, bawah) dalam mode tabel dengan target RDBMS. Misalnya, AWS DMS tidak mendukung memiliki dua koleksi bernama `Field1` dan `field1`.
- Mode tabel dan mode dokumen memiliki batasan yang dijelaskan sebelumnya.
- Migrasi secara paralel menggunakan autosegmentasi memiliki keterbatasan yang dijelaskan sebelumnya.
- Filter sumber tidak didukung untuk MongoDB.
- AWS DMS tidak mendukung dokumen di mana tingkat bersarang lebih besar dari 97.
- AWS DMS tidak mendukung fitur MongoDB versi 5.0 berikut:
 - Resharding langsung
 - Enkripsi Tingkat Bidang Sisi Klien (CSFLE)
 - Migrasi koleksi Timeseries

Note

Koleksi rangkaian waktu yang dimigrasikan dalam fase pemuatan penuh akan dikonversi ke koleksi normal di Amazon DocumentDB, karena DocumentDB tidak mendukung koleksi rangkaian waktu.

Pengaturan konfigurasi titik akhir saat menggunakan MongoDB sebagai sumber untuk AWS DMS

Ketika Anda mengatur titik akhir sumber MongoDB Anda, Anda dapat menentukan beberapa pengaturan konfigurasi titik akhir menggunakan konsol AWS DMS.

Tabel berikut menjelaskan pengaturan konfigurasi yang tersedia ketika menggunakan basis data MongoDB sebagai sumber AWS DMS.

Pengaturan (atribut)	Nilai valid	Nilai default dan deskripsi
Modus otentikasi	"none" "password"	Nilai "password" meminta nama pengguna dan kata sandi. Saat "none" ditentukan, parameter nama pengguna dan kata sandi tidak digunakan.
Sumber otentikasi	Sebuah nama basis data MongoDB yang valid.	Nama basis data MongoDB yang ingin Anda gunakan untuk melakukan validasi Anda untuk autentikasi. Nilai default-nya adalah "admin".
Mekanisme otentikasi	"default" "mongodb_cr" "scram_sha_1"	Mekanisme otentikasi. Nilai "default" adalah "scram_sha_1". Pengaturan ini tidak digunakan saat authType diatur menjadi "no".
Modus metadata	Dokumen dan tabel	Memilih mode dokumen atau mode tabel.
Jumlah dokumen yang akan dipindai (docsToInvestigate)	Sebuah integer positif lebih besar dari 0.	Gunakan opsi ini dalam mode tabel hanya untuk menentukan definisi tabel target.
_id sebagai kolom terpisah	Centang tanda di kotak	Kotak tanda centang opsional yang membuat kolom kedua bernama _id yang bertindak sebagai kunci utama.

Pengaturan (atribut)	Nilai valid	Nilai default dan deskripsi
socketTimeoutMS	ANGKA Atribut Koneksi Ekstra (ECA) saja.	Pengaturan ini dalam satuan milidetik dan mengonfigurasi batas waktu koneksi untuk klien MongoDB. Jika nilainya kurang dari atau sama dengan nol, maka default klien MongoDB digunakan.
UseUpdateLookUp	boolean true false	Jika benar, selama peristiwa pembaruan CDCAWS DMS, salin seluruh dokumen yang diperbarui ke target. Saat disetel ke false, AWS DMS gunakan perintah pembaruan MongoDB untuk hanya memperbarui bidang yang dimodifikasi dalam dokumen pada target.
ReplicateShardCollections	boolean true false	Jika benar, AWS DMS mereplikasi data ke koleksi pecahan. AWS DMS hanya menggunakan pengaturan ini jika titik akhir target adalah cluster elastis DocumentDB. Jika pengaturan ini benar, perhatikan hal berikut: <ul style="list-style-type: none"> • Anda harus mengatur TargetTablePrepMode ke nothing. • AWS DMS secara otomatis disetel useUpdateLookUp ke false.

Jika Anda memilih Dokumen sebagai Mode metadata, opsi yang berbeda akan tersedia.

Jika titik akhir target adalah DocumentDB, pastikan untuk menjalankan migrasi dalam mode Dokumen Juga, ubah titik akhir sumber Anda dan pilih opsi `_id` sebagai kolom terpisah. Ini adalah prasyarat wajib jika beban kerja MongoDB sumber Anda melibatkan transaksi.

Jenis data sumber untuk MongoDB

Migrasi data yang menggunakan MongoDB sebagai sumber untuk AWS DMS mendukung sebagian besar jenis data MongoDB. Pada tabel berikut, Anda dapat menemukan jenis data sumber MongoDB yang didukung saat menggunakan AWS DMS dan pemetaan default dari jenis data AWS DMS. Untuk informasi lebih lanjut tentang jenis data MongoDB, lihat [jenis BSON](#) dalam dokumentasi MongoDB.

Untuk informasi tentang cara melihat jenis data yang dipetakan dalam target, lihat bagian titik akhir target yang Anda gunakan.

Untuk informasi tambahan tentang jenis data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Jenis data MongoDB	Jenis data AWS DMS
Boolean	Bool
Binary	BLOB
Date	Date
Timestamp	Date
Int	INT4
Long	INT8
Double	REAL8
String (UTF-8)	CLOB
Array	CLOB
OID	String
REGEX	CLOB
CODE	CLOB

Menggunakan Amazon DocumentDB (dengan kompatibilitas MongoDB) sebagai sumber untuk AWS DMS

Untuk informasi tentang versi Amazon DocumentDB (dengan kompatibilitas MongoDB) AWS DMS yang mendukung sebagai sumber, lihat [Sumber untuk AWS DMS](#)

Dengan menggunakan Amazon DocumentDB sebagai sumber, Anda dapat memigrasikan data dari satu klaster Amazon DocumentDB ke klaster Amazon DocumentDB lain. Anda juga dapat memigrasi

data dari klaster Amazon DocumentDB ke salah satu titik akhir target lainnya yang didukung oleh AWS DMS.

Jika Anda baru mengenal Amazon DocumentDB, perhatikan konsep penting berikut untuk database Amazon DocumentDB:

- Catatan di Amazon DocumentDB adalah dokumen, struktur data yang terdiri dari pasangan bidang dan nilai. Nilai bidang dapat mencakup dokumen, rangkaian, dan rangkaian dokumen lain. Dokumen kurang lebih setara dengan baris dalam tabel basis data relasional.
- Koleksi di Amazon DocumentDB adalah sekelompok dokumen, dan kira-kira setara dengan tabel database relasional.
- Database di Amazon DocumentDB adalah sekumpulan koleksi, dan kira-kira setara dengan skema dalam database relasional.

AWS DMS mendukung dua mode migrasi saat menggunakan Amazon DocumentDB sebagai sumber, mode dokumen, dan mode tabel. Anda menentukan mode migrasi saat membuat titik akhir sumber Amazon DocumentDB di konsol, menggunakan AWS DMS opsi mode Metadata atau atribut koneksi tambahan. `nestingLevel` Berikut, Anda dapat menemukan penjelasan bagaimana pilihan mode migrasi memengaruhi format yang dihasilkan dari data target.

Mode dokumen

Pada mode dokumen, dokumen JSON dimigrasi apa adanya. Itu artinya data dokumen dikonsolidasikan menjadi salah satu dari dua item. Saat Anda menggunakan basis data relasional sebagai target, data merupakan satu kolom bernama `_doc` dalam tabel target. Ketika Anda menggunakan basis data nonrelasional sebagai target, data merupakan dokumen JSON tunggal. Mode dokumen adalah mode default, yang kami sarankan ketika bermigrasi ke target Amazon DocumentDB.

Misalnya, pertimbangkan dokumen-dokumen berikut dalam koleksi Amazon DocumentDB yang disebut `myCollection`

```
> db.myCollection.find()
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe0"), "a" : 1, "b" : 2, "c" : 3 }
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe1"), "a" : 4, "b" : 5, "c" : 6 }
```

Setelah memigrasikan data ke tabel basis data relasional menggunakan mode dokumen, data terstruktur sebagai berikut. Bidang data dalam dokumen dikonsolidasikan ke kolom `_doc`.

oid_id	_doc
5a94815f40bd44d1b02bdfe0	{ "a" : 1, "b" : 2, "c" : 3 }
5a94815f40bd44d1b02bdfe1	{ "a" : 4, "b" : 5, "c" : 6 }

Anda dapat mengatur atribut sambungan tambahan `extractDocID` secara fakultatif ke `true` untuk membuat kolom kedua bernama `"_id"` yang bertindak sebagai kunci utama. Jika Anda akan menggunakan `change data capture (CDC)`, atur parameter ini menjadi `true` kecuali saat menggunakan Amazon DocumentDB sebagai target.

Note

Jika Anda menambahkan koleksi baru ke basis data sumber, AWS DMS menciptakan tabel target baru untuk koleksi dan mereplikasi dokumen apa pun.

Mode tabel

Dalam mode tabel, AWS DMS mengubah setiap bidang tingkat atas dalam dokumen Amazon DocumentDB menjadi kolom di tabel target. Jika bidang bersarang, AWS DMS meratakan nilai-nilai bersarang ke dalam satu kolom. AWS DMS kemudian menambahkan bidang kunci dan jenis data untuk tabel target rangkaian kolom.

Untuk setiap dokumen Amazon DocumentDB AWS DMS, tambahkan setiap kunci dan ketik ke set kolom tabel target. Misalnya, dengan menggunakan mode tabel, AWS DMS memigrasikan contoh sebelumnya ke dalam tabel berikut.

oid_id	a	b	c
5a94815f40bd44d1b02bdfe0	1	2	3
5a94815f40bd44d1b02bdfe1	4	5	6

Nilai-nilai bersarang diratakan menjadi sebuah kolom yang berisi nama-nama kunci yang dipisahkan oleh titik. Kolom dinamakan menggunakan gabungan nama bidang yang diratakan yang dipisahkan oleh titik. Misalnya, AWS DMS memigrasikan dokumen JSON dengan bidang nilai-nilai bersarang seperti `{"a" : {"b" : {"c": 1}}}` ke dalam sebuah kolom bernama `a.b.c`.

Untuk membuat kolom target, AWS DMS memindai sejumlah dokumen Amazon DocumentDB tertentu dan membuat satu set semua bidang dan jenisnya. AWS DMS kemudian menggunakan set ini untuk membuat kolom dari tabel target. Jika Anda membuat atau memodifikasi titik akhir sumber Amazon DocumentDB menggunakan konsol, Anda dapat menentukan jumlah dokumen yang akan dipindai. Nilai default adalah 1.000 dokumen. Jika Anda menggunakan AWS CLI, Anda dapat menggunakan atribut sambungan tambahan `docsToInvestigate`.

Dalam mode tabel, AWS DMS mengelola dokumen dan koleksi seperti ini:

- Ketika Anda menambahkan dokumen ke koleksi yang ada, dokumen itu direplikasi. Jika ada bidang yang tidak ada dalam target, bidang tersebut tidak direplikasi.
- Ketika Anda memperbarui dokumen, dokumen yang diperbarui tersebut direplikasi. Jika ada bidang yang tidak ada dalam target, bidang tersebut tidak direplikasi.
- Menghapus dokumen sepenuhnya didukung.
- Menambahkan koleksi baru tidak menghasilkan tabel baru pada target ketika dilakukan selama tugas CDC.
- Pada fase Change Data Capture (CDC), AWS DMS tidak mendukung penggantian nama koleksi.

Topik

- [Menetapkan izin untuk menggunakan Amazon DocumentDB sebagai sumber](#)
- [Mengonfigurasi CDC untuk kluster Amazon DocumentDB](#)
- [Menghubungkan ke Amazon DocumentDB menggunakan TLS](#)
- [Membuat titik akhir sumber Amazon DocumentDB](#)
- [Melakukan segmentasi koleksi Amazon DocumentDB dan migrasi secara paralel](#)
- [Migrasi beberapa database saat menggunakan Amazon DocumentDB sebagai sumber AWS DMS](#)
- [Keterbatasan saat menggunakan Amazon DocumentDB sebagai sumber AWS DMS](#)
- [Menggunakan pengaturan titik akhir dengan Amazon DocumentDB sebagai sumber](#)

- [Tipe data sumber untuk Amazon DocumentDB](#)

Menetapkan izin untuk menggunakan Amazon DocumentDB sebagai sumber

Ketika menggunakan sumber Amazon DocumentDB untuk migrasi AWS DMS, Anda dapat membuat akun pengguna dengan hak istimewa root. Atau Anda dapat membuat pengguna dengan izin hanya untuk basis data yang akan dimigrasi.

Kode berikut membuat pengguna sebagai akun root.

```
use admin
db.createUser(
  {
    user: "root",
    pwd: "password",
    roles: [ { role: "root", db: "admin" } ]
  })
```

Untuk Amazon DocumentDB 3.6, kode berikut membuat pengguna dengan hak istimewa minimal pada database yang akan dimigrasi.

```
use database_to_migrate
db.createUser(
  {
    user: "dms-user",
    pwd: "password",
    roles: [ { role: "read", db: "db_name" }, "read" ]
  })
```

Untuk Amazon DocumentDB 4.0 dan yang lebih tinggi AWS DMS, menggunakan aliran perubahan di seluruh penerapan. Di sini, kode berikut membuat pengguna dengan hak istimewa minimal.

```
db.createUser(
  {
    user: "dms-user",
    pwd: "password",
    roles: [ { role: "readAnyDatabase", db: "admin" } ]
  })
```

Mengonfigurasi CDC untuk klaster Amazon DocumentDB

Untuk menggunakan replikasi atau CDC yang sedang berlangsung dengan Amazon DocumentDBAWS DMS, memerlukan akses ke aliran perubahan klaster Amazon DocumentDB. Untuk deskripsi urutan yang diatur waktu dari peristiwa pembaruan di koleksi dan basis data klaster Anda, lihat [Menggunakan aliran perubahan](#) dalam Panduan Developer Amazon DocumentDB.

Lakukan autentikasi klaster Amazon DocumentDB Anda menggunakan shell MongoDB. Kemudian jalankan perintah berikut untuk mengaktifkan perubahan aliran.

```
db.adminCommand({modifyChangeStreams: 1,
  database: "DB_NAME",
  collection: "",
  enable: true});
```

Pendekatan ini memungkinkan aliran perubahan untuk semua koleksi dalam basis data Anda. Setelah aliran perubahan diaktifkan, Anda dapat membuat tugas migrasi yang memigrasi data yang ada dan pada saat yang sama mereplikasi perubahan yang sedang berlangsung. AWS DMS terus menangkap dan menerapkan perubahan bahkan setelah data massal dimuat. Akhirnya, sumber dan target basis data menyinkronkan, meminimalkan downtime untuk migrasi.

Note

AWS DMS menggunakan log operasi (oplog) untuk menangkap perubahan selama replikasi yang sedang berlangsung. Jika Amazon DocumentDB menghapus catatan dari oplog AWS DMS sebelum membacanya, tugas Anda akan gagal. Kami merekomendasikan ukuran oplog untuk mempertahankan perubahan setidaknya selama 24 jam.

Menghubungkan ke Amazon DocumentDB menggunakan TLS

Secara default, klaster Amazon DocumentDB baru dibuat menerima sambungan aman hanya menggunakan Keamanan Lapisan Pengangkutan (TLS). Ketika TLS diaktifkan, setiap sambungan ke Amazon DocumentDB memerlukan kunci publik.

Anda dapat mengambil kunci publik untuk Amazon DocumentDB dengan mengunduh file `rds-combined-ca-bundle.pem` dari Amazon S3 bucket yang di-host oleh AWS. Untuk informasi lebih

lanjut tentang mengunduh file ini, lihat [Mengenripsi sambungan menggunakan TLS](#) dalam Panduan Developer Amazon DocumentDB.

Setelah Anda mengunduh file `rds-combined-ca-bundle.pem`, Anda dapat mengimpor kunci publik yang berisi ke AWS DMS. Langkah-langkah berikut menjelaskan cara melakukannya.

Untuk mengimpor kunci publik Anda menggunakan konsol AWS DMS

1. Masuklah ke AWS Management Console dan pilih AWS DMS.
2. Pilih Sertifikat di panel navigasi.
3. Pilih Impor sertifikat. Laman Impor sertifikat CA baru akan muncul.
4. Pada bagian Konfigurasi sertifikat, lakukan salah satu hal berikut:
 - Untuk Pengenal sertifikat, masukkan nama unik untuk sertifikat tersebut, seperti `docdb-cert`.
 - Pilih Pilih berkas, navigasikan ke lokasi tempat Anda menyimpan `rds-combined-ca-bundle.pem`, dan pilih itu.
5. Pilih Tambahkan sertifikat CA baru.

AWS CLIContoh berikut ini menggunakan perintah AWS DMS `import-certificate` untuk mengimpor kunci publik file `rds-combined-ca-bundle.pem`.

```
aws dms import-certificate \  
  --certificate-identifier docdb-cert \  
  --certificate-pem file:///./rds-combined-ca-bundle.pem
```

Membuat titik akhir sumber Amazon DocumentDB

Anda dapat membuat titik akhir sumber Amazon DocumentDB menggunakan konsol atau AWS CLI. Gunakan prosedur berikut dengan konsol.

Mengonfigurasi titik akhir sumber Amazon DocumentDB menggunakan konsol AWS DMS

1. Masuklah ke AWS Management Console dan pilih AWS DMS.
2. Pilih Titik akhir dari panel navigasi, lalu pilih Buat titik akhir.
3. Untuk Pengidentifikasi titik akhir, berikan nama yang membantu Anda mengidentifikasinya dengan mudah, seperti `docdb-source`.
4. Untuk Mesin sumber, pilih Amazon DocumentDB (dengan kompatibilitas MongoDB).

5. Untuk Nama server, masukkan nama server di lokasi titik akhir basis data Amazon DocumentDB Anda berada. Misalnya, Anda dapat memasukkan nama DNS publik dari instans Amazon EC2 Anda, seperti `democluster.cluster-cjf6q8nxfefi.us-east-2.docdb.amazonaws.com`.
6. Untuk Port, masukkan 27017.
7. Untuk Mode SSL, pilih verifikasi penuh. Jika Anda telah menonaktifkan SSL pada kluster Amazon DocumentDB Anda, Anda dapat melewati langkah ini.
8. Untuk Sertifikat CA, pilih sertifikat Amazon DocumentDB, `rdc-combined-ca-bundle.pem`. Untuk petunjuk tentang cara menambahkan sertifikat ini, lihat [Menghubungkan ke Amazon DocumentDB menggunakan TLS](#).
9. Untuk Nama basis data, masukkan nama dari basis data yang akan dimigrasi.

Gunakan prosedur berikut dengan CLI.

Mengonfigurasi titik akhir sumber Amazon DocumentDB menggunakan AWS CLI

- Jalankan perintah AWS DMS `create-endpoint` berikut untuk mengonfigurasi titik akhir sumber Amazon DocumentDB, menggantikan placeholder dengan nilai-nilai Anda sendiri.

```
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
```

Melakukan segmentasi koleksi Amazon DocumentDB dan migrasi secara paralel

Untuk meningkatkan kinerja tugas migrasi, titik akhir sumber Amazon DocumentDB mendukung dua opsi fitur beban penuh paralel dalam pemetaan tabel. Dengan kata lain, Anda dapat memigrasikan koleksi secara paralel dengan menggunakan opsi autosegmentasi atau segmentasi rentang pemetaan tabel untuk beban penuh paralel dalam pengaturan JSON. Opsi segmentasi otomatis memungkinkan Anda menentukan kriteria untuk AWS DMS secara otomatis mengelompokkan

sumber Anda untuk migrasi di setiap utas. Opsi segmentasi rentang memungkinkan Anda memberi tahu AWS DMS rentang spesifik setiap segmen agar DMS dapat bermigrasi di setiap utas. Untuk informasi selengkapnya tentang pengaturan, lihat [Tabel dan koleksi pengaturan aturan dan operasi](#).

Migrasi database Amazon DocumentDB secara paralel menggunakan rentang autosegmentasi

Anda dapat memigrasikan dokumen secara paralel dengan menentukan kriteria AWS DMS untuk secara otomatis membagi (mengelompokkan) data Anda untuk setiap utas, terutama jumlah dokumen yang akan dimigrasi per utas. Dengan menggunakan pendekatan ini, AWS DMS upaya untuk mengoptimalkan batas segmen untuk kinerja maksimum per utas.

Anda dapat menentukan kriteria segmentasi menggunakan opsi pengaturan tabel berikut dalam pemetaan tabel:

Opsi tabel-pengaturan	Deskripsi
"type"	(Wajib) Setel ke "partitions-auto" Amazon DocumentDB sebagai sumber.
"number-of-partitions"	(Opsional) Jumlah total partisi (segmen) yang digunakan untuk migrasi. Defaultnya adalah 16.
"collection-count-from-meta data"	(Opsional) Jika diatur ke <code>true</code> , AWS DMS menggunakan perkiraan jumlah koleksi untuk menentukan jumlah partisi. Jika disetel ke <code>false</code> , AWS DMS gunakan jumlah koleksi yang sebenarnya. Default-nya adalah <code>true</code> .
"max-records-skip-per-page"	(Opsional) Jumlah catatan yang harus dilewati sekaligus saat menentukan batas untuk setiap partisi. AWS DMS menggunakan pendekatan <code>skip paginated</code> untuk menentukan batas minimum untuk partisi. Defaultnya adalah 10000. Menyetel nilai yang relatif besar dapat mengakibatkan batas waktu kurser dan kegagalan tugas. Menetapkan nilai yang relatif rendah menghasilkan lebih banyak operasi per halaman dan beban penuh yang lebih lambat.

Opsi tabel-pengaturan	Deskripsi
"batch-size"	(Opsional) Membatasi jumlah dokumen yang dikembalikan dalam satu batch. Setiap batch membutuhkan perjalanan pulang pergi ke server. Jika ukuran batch nol (0), kursor menggunakan ukuran batch maksimum yang ditentukan server. Default-nya adalah 0.

Contoh berikut menunjukkan pemetaan tabel untuk autosegmentasi.

```
{
  "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
      }
    }
  ]
}
```

}

Segmentasi otomatis memiliki batasan berikut. Migrasi untuk setiap segmen mengambil jumlah koleksi dan minimum `_id` untuk koleksi secara terpisah. Kemudian menggunakan lompatan paginasi untuk menghitung batas minimum untuk segmen itu. Oleh karena itu, pastikan bahwa `_id` nilai minimum untuk setiap koleksi tetap konstan sampai semua batas segmen dalam koleksi dihitung. Jika Anda mengubah `_id` nilai minimum untuk koleksi selama perhitungan batas segmennya, ini dapat menyebabkan kehilangan data atau kesalahan baris duplikat.

Migrasi database Amazon DocumentDB secara paralel menggunakan rentang segmen tertentu

Contoh berikut menunjukkan koleksi Amazon DocumentDB yang memiliki tujuh item, `_id` dan sebagai kunci utama.

Key	Value	Type
▼ (1) ObjectId("5f805c74873173399a278d78")	{ 3 fields }	Object
<input type="checkbox"/> _id	ObjectId("5f805c74873173399a278d78")	ObjectId
<input checked="" type="checkbox"/> num	1	Int32
<input type="checkbox"/> name	a	String
▼ (2) ObjectId("5f805c97873173399a278d79")	{ 3 fields }	Object
<input type="checkbox"/> _id	ObjectId("5f805c97873173399a278d79")	ObjectId
<input checked="" type="checkbox"/> num	2	Int32
<input type="checkbox"/> name	b	String
▼ (3) ObjectId("5f805cb0873173399a278d7a")	{ 3 fields }	Object
<input type="checkbox"/> _id	ObjectId("5f805cb0873173399a278d7a")	ObjectId
<input checked="" type="checkbox"/> num	3	Int32
<input type="checkbox"/> name	c	String
▼ (4) ObjectId("5f805cbb873173399a278d7b")	{ 3 fields }	Object
<input type="checkbox"/> _id	ObjectId("5f805cbb873173399a278d7b")	ObjectId
<input checked="" type="checkbox"/> num	4	Int32
<input type="checkbox"/> name	d	String
▼ (5) ObjectId("5f805cc5873173399a278d7c")	{ 3 fields }	Object
<input type="checkbox"/> _id	ObjectId("5f805cc5873173399a278d7c")	ObjectId
<input checked="" type="checkbox"/> num	5	Int32
<input type="checkbox"/> name	e	String
▼ (6) ObjectId("5f805cd0873173399a278d7d")	{ 3 fields }	Object
<input type="checkbox"/> _id	ObjectId("5f805cd0873173399a278d7d")	ObjectId
<input checked="" type="checkbox"/> num	6	Int32
<input type="checkbox"/> name	f	String
▼ (7) ObjectId("5f805cdd873173399a278d7e")	{ 3 fields }	Object
<input type="checkbox"/> _id	ObjectId("5f805cdd873173399a278d7e")	ObjectId
<input checked="" type="checkbox"/> num	7	Int32
<input type="checkbox"/> name	g	String

Untuk membagi koleksi menjadi tiga segmen dan bermigrasi secara paralel, Anda dapat menambahkan aturan pemetaan tabel untuk tugas migrasi Anda seperti yang ditunjukkan dalam contoh JSON berikut.

```
{ // 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

```

Bahwa definisi pemetaan tabel membagi koleksi sumber menjadi tiga segmen dan bermigrasi secara paralel. Berikut ini adalah batas-batas segmentasi.

```

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)

```

Setelah tugas migrasi selesai, Anda dapat melakukan verifikasi dari log tugas yang dimuat tabel secara paralel, seperti yang ditunjukkan dalam contoh berikut. Anda juga dapat memverifikasi klausa Amazon find DocumentDB yang digunakan untuk membongkar setiap segmen dari tabel sumber.

```

[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
transferred 480.

[TASK_MANAGER    ] I: Load finished for segment #1 of table
'testdatabase'. 'testtable' (Id = 1) by subtask 1. 2 records transferred.
```

Saat ini, AWS DMS mendukung tipe data Amazon DocumentDB berikut sebagai kolom kunci segmen:

- Dua Kali
- String
- ObjectId
- 32 bit integer
- 64 bit integer

Migrasi beberapa database saat menggunakan Amazon DocumentDB sebagai sumber AWS DMS

AWS DMS versi 3.4.5 dan yang lebih tinggi mendukung migrasi beberapa database dalam satu tugas hanya untuk Amazon DocumentDB versi 4.0 dan yang lebih tinggi. Jika Anda ingin memigrasi beberapa database, lakukan hal berikut:

1. Saat Anda membuat titik akhir sumber Amazon DocumentDB:

- Dalam AWS Management Console for AWS DMS, biarkan nama Database kosong di bawah konfigurasi Endpoint pada halaman Create endpoint.
 - Di AWS Command Line Interface (AWS CLI), tetapkan nilai string kosong ke DatabaseNameparameter di DocumentDbSettings yang Anda tentukan untuk tindakan tersebut. CreateEndpoint
2. Untuk setiap database yang ingin Anda migrasi dari titik akhir sumber Amazon DocumentDB ini, tentukan nama setiap database sebagai nama skema dalam pemetaan tabel untuk tugas menggunakan input terpandu di konsol atau langsung di JSON. Untuk informasi lebih lanjut tentang input yang dipandu, lihat deskripsi [Menentukan pemilihan tabel dan transformasi aturan dari konsol](#). Untuk informasi lebih lanjut tentang JSON, lihat [Aturan seleksi dan tindakan](#).

Misalnya, Anda dapat menentukan JSON berikut untuk memigrasikan tiga database Amazon DocumentDB.

Example Memigrasi semua tabel dalam skema

JSON berikut memigrasikan semua tabel dari Customers, Orders, dan Suppliers database di sumber Anda endpoint ke titik akhir target Anda.

```
{
  "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"
  ]
}
```

```
}
```

Keterbatasan saat menggunakan Amazon DocumentDB sebagai sumber AWS DMS

Berikut ini adalah keterbatasan ketika menggunakan Amazon DocumentDB sebagai sumber untuk AWS DMS:

- Saat opsi `_id` diatur sebagai kolom terpisah, string ID tidak dapat melebihi 200 karakter.
- ID objek dan kunci jenis rangkaian dikonversi ke kolom yang diawali dengan `oid` dan `array` dalam mode tabel.

Secara internal, kolom ini diacu dengan nama prefiks. Jika Anda menggunakan aturan transformasi di AWS DMS yang mengacu kolom ini, pastikan untuk menentukan kolom prefiks. Sebagai contoh, tentukan `$_oid__id` dan bukan `$_id`, atau `$_array__addresses` dan bukan `$_addresses`.

- Nama koleksi dan nama kunci tidak dapat menyertakan simbol dolar (\$).
- Mode tabel dan mode dokumen memiliki keterbatasan seperti yang dibahas sebelumnya.
- Migrasi secara paralel menggunakan autosegmentasi memiliki keterbatasan yang dijelaskan sebelumnya.
- Sumber Amazon DocumentDB (kompatibel dengan MongoDB) tidak mendukung penggunaan stempel waktu tertentu sebagai posisi awal untuk pengambilan data perubahan (CDC). Tugas replikasi yang sedang berlangsung mulai menangkap perubahan terlepas dari timestamp.
- Bila menggunakan DocumentDB (MongoDB kompatibel) sebagai sumber, DMS dapat menangani maksimal 250 record per detik.
- AWS DMS tidak mendukung dokumen di mana tingkat bersarang lebih besar dari 97.
- Filter sumber tidak didukung untuk DocumentDB.
- AWS DMS tidak mendukung replikasi CDC (ubah pengambilan data) untuk DocumentDB sebagai sumber dalam mode cluster elastis.

Menggunakan pengaturan titik akhir dengan Amazon DocumentDB sebagai sumber

Anda dapat menggunakan pengaturan titik akhir untuk mengonfigurasi basis data sumber Amazon DocumentDB mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir sumber menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--doc-db-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan titik akhir yang dapat Anda gunakan dengan Amazon DocumentDB sebagai sumber.

Nama atribut	Nilai valid	Nilai default dan deskripsi
NestingLevel	"none" "one"	"none" — Tentukan "none" untuk menggunakan mode dokumen. Tentukan "one" untuk menggunakan mode tabel.
ExtractDocID	boolean true false	false — Gunakan atribut ini ketika NestingLevel diatur menjadi "none". Jika database target Anda adalah Amazon DocumentDB, atur. '{"ExtractDocID": true}'
DocsToInvestigate	Sebuah integer positif lebih besar dari 0.	1000 — Gunakan atribut ini ketika NestingLevel diatur menjadi "one".
ReplicateShardCollections	boolean true false	Jika benar, AWS DMS mereplikasi data ke koleksi pecahan. AWS DMS hanya menggunakan pengaturan ini jika titik akhir target adalah cluster elastis DocumentDB. Jika pengaturan ini benar, perhatikan hal berikut: <ul style="list-style-type: none"> • Anda harus mengatur TargetTablePrepMode ke nothing. • AWS DMS secara otomatis disetel useUpdateLookup ke false.

Tipe data sumber untuk Amazon DocumentDB

Dalam tabel berikut, Anda dapat menemukan tipe data sumber Amazon DocumentDB yang didukung saat menggunakan AWS DMS. Anda juga dapat menemukan pemetaan default dari jenis data AWS DMS dalam tabel ini. Untuk informasi lebih lanjut tentang jenis data, lihat [Jenis BSON](#) dalam dokumentasi MongoDB.

Untuk informasi tentang cara melihat jenis data yang dipetakan dalam target, lihat bagian titik akhir target yang Anda gunakan.

Untuk informasi tambahan tentang jenis data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Tipe data Amazon DocumentDB	Jenis data AWS DMS
Boolean	Bool
Binary	BLOB
Date	Date
Timestamp	Date
Int	INT4
Long	INT8
Double	REAL8
String (UTF-8)	CLOB
Array	CLOB
OID	String

Menggunakan Amazon S3 sebagai sumber untuk AWS DMS

Anda dapat memigrasikan data dari Amazon S3 bucket menggunakan AWS DMS. Untuk melakukan ini, berikan akses ke Amazon S3 bucket yang berisi file data satu atau lebih. Dalam bucket S3, termasuk file JSON yang menggambarkan pemetaan antara data dan tabel basis data data dalam file-file tersebut.

File data sumber harus ada dalam Amazon S3 bucket sebelum beban penuh dimulai. Anda menentukan nama bucket menggunakan parameter `bucketName`.

File data sumber harus dalam format comma-separated value (.csv). Nama mereka menggunakan konvensi penamaan berikut. Dalam konvensi ini, *schemaName* adalah skema sumber dan *tableName* adalah nama tabel dalam skema itu.

```
/schemaName/tableName/LOAD001.csv  
/schemaName/tableName/LOAD002.csv  
/schemaName/tableName/LOAD003.csv  
...
```

Misalnya, anggaplah file data Anda berada di mybucket, di jalur Amazon S3 berikut.

```
s3://mybucket/hr/employee
```

Pada waktu unggah, AWS DMS mengasumsikan bahwa nama skema sumber adalah hr, dan bahwa nama tabel sumber adalah employee.

Selain bucketName (diperlukan), Anda dapat memberikan parameter bucketFolder untuk menentukan di mana AWS DMS harus mencari file data dalam Amazon S3 bucket. Melanjutkan contoh sebelumnya, jika Anda mengatur bucketFolder menjadi sourcedata, maka AWS DMS membaca file data di jalur berikut.

```
s3://mybucket/sourcedata/hr/employee
```

Anda dapat menentukan pembatas kolom, baris pembatas, indikator nilai null, dan parameter lainnya menggunakan atribut sambungan tambahan. Untuk informasi selengkapnya, lihat [Pengaturan titik akhir untuk Amazon S3 sebagai sumber AWS DMS](#).

Anda dapat menentukan pemilik bucket dan mencegah sniping dengan menggunakan ExpectedBucketOwner Pengaturan titik akhir Amazon S3, seperti yang ditunjukkan berikut. Kemudian, ketika Anda membuat permintaan untuk menguji sambungan atau melakukan migrasi, S3 memeriksa ID akun pemilik bucket terhadap parameter yang ditentukan.

```
--s3-settings='{"ExpectedBucketOwner": "AWS_Account_ID"}'
```

Topik

- [Mendefinisikan tabel eksternal untuk Amazon S3 sebagai sumber untuk AWS DMS](#)
- [Menggunakan CDC dengan Amazon S3 sebagai sumber untuk AWS DMS](#)
- [Prasyarat saat menggunakan Amazon S3 sebagai sumber untuk AWS DMS](#)

- [Batasan saat menggunakan Amazon S3 sebagai sumber AWS DMS](#)
- [Pengaturan titik akhir untuk Amazon S3 sebagai sumber AWS DMS](#)
- [Jenis data sumber untuk Amazon S3](#)

Mendefinisikan tabel eksternal untuk Amazon S3 sebagai sumber untuk AWS DMS

Selain file data, Anda juga harus memberikan definisi tabel eksternal. Definisi tabel eksternal adalah dokumen JSON yang menjelaskan bagaimana AWS DMS harus menafsirkan data dari Amazon S3. Ukuran maksimum dokumen ini adalah 2 MB. Jika Anda membuat titik akhir sumber menggunakan AWS DMS Konsol Manajemen, Anda dapat memasukkan JSON langsung ke kotak pemetaan table. Jika Anda menggunakan API AWS Command Line Interface (AWS CLI) atau AWS DMS untuk melakukan migrasi, Anda dapat membuat file JSON untuk menentukan definisi tabel eksternal.

Misalkan Anda memiliki file data yang mencakup hal berikut.

```
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
```

Berikut ini adalah contoh definisi tabel eksternal untuk data ini.

```
{
  "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"
}
]
}

```

Unsur-unsur dalam dokumen JSON ini adalah sebagai berikut:

TableCount — jumlah tabel sumber. Dalam contoh ini, hanya ada satu tabel.

Tables — susunan yang terdiri dari satu peta JSON per tabel sumber. Dalam contoh ini, hanya ada satu peta. Setiap peta terdiri atas elemen-elemen berikut:

- **TableName** — nama tabel sumber.
- **TablePath** — jalur di Amazon S3 bucket Anda di mana AWS DMS dapat menemukan file beban data lengkap. Jika `bucketFolder` nilai yang ditentukan, nilainya adalah prepended ke jalan.
- **TableOwner** — nama skema untuk tabel ini.
- **TableColumns** — susunan dari satu peta atau lebih, masing-masing menggambarkan kolom dalam tabel sumber:
 - **ColumnName** — nama kolom dalam tabel sumber.
 - **ColumnType** — jenis data untuk kolom. Untuk jenis data yang valid, lihat [Jenis data sumber untuk Amazon S3](#).
 - **ColumnLength** — jumlah byte dalam kolom ini. Panjang kolom maksimum dibatasi untuk 2147483647 Bytes (2.047 MegaBytes) karena sumber S3 tidak mendukung mode LOB LENGKAP. `ColumnLength` berlaku untuk tipe data berikut:

- `BYTE`
- `STRING`
- `ColumnNullable` — nilai Boolean yang `true` jika kolom ini dapat berisi nilai-nilai `NULL` (`default=false`).
- `ColumnIsPk` — nilai Boolean yang `true` jika kolom ini adalah bagian dari kunci primer (`default=false`).
- `ColumnDateFormat` - Format tanggal masukan untuk kolom dengan tipe `DATE`, `TIME`, dan `DATETIME`, dan digunakan untuk mengurai string data menjadi objek tanggal. Nilai yang mungkin termasuk:
 - `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` — jumlah total kolom. Nomor ini harus sesuai dengan jumlah elemen dalam susunan `TableColumns`.

Jika Anda tidak menentukan sebaliknya, AWS DMS mengasumsikan bahwa `ColumnLength` adalah nol.

Note

Dalam versi yang didukung AWS DMS, data sumber S3 juga dapat berisi kolom operasi opsional sebagai kolom pertama sebelum nilai kolom `TableName`. Kolom operasi ini mengidentifikasi operasi (`INSERT`) yang digunakan untuk memigrasikan data ke titik akhir target S3 selama beban penuh.

Jika ada, nilai kolom ini adalah karakter awal dari operasi kata kunci `INSERT (I)`. Jika ditentukan, kolom ini umumnya menunjukkan bahwa sumber S3 dibuat oleh DMS sebagai target S3 selama migrasi sebelumnya.

Dalam versi DMS sebelum 3.4.2, kolom ini tidak ada dalam data sumber S3 yang dibuat dari beban penuh DMS sebelumnya. Menambahkan kolom ini ke data target S3 memungkinkan format semua baris ditulis ke target S3 agar konsisten apakah mereka ditulis selama beban

penuh atau selama beban CDC. Untuk informasi lebih lanjut tentang pilihan format S3 target data, lihat [Menunjukkan operasi sumber DB dalam data S3 bermigrasi](#).

Untuk kolom dari jenis NUMERIC, tentukan presisi dan skala. Presisi adalah jumlah digit dalam angka, dan skala adalah jumlah digit di sebelah kanan titik desimal. Anda menggunakan unsur-unsur `ColumnPrecision` dan `ColumnScale` untuk ini, seperti yang ditunjukkan berikut.

```
...
{
  "ColumnName": "HourlyRate",
  "ColumnType": "NUMERIC",
  "ColumnPrecision": "5"
  "ColumnScale": "2"
}
...
```

Untuk kolom tipe DATETIME dengan data yang berisi detik pecahan, tentukan skalanya. Skala adalah jumlah digit untuk detik fraksional, dan dapat berkisar dari 0 sampai 9. Anda menggunakan `ColumnScale` elemen untuk ini, seperti yang ditunjukkan berikut.

```
...
{
  "ColumnName": "HireDate",
  "ColumnType": "DATETIME",
  "ColumnScale": "3"
}
...
```

Jika Anda tidak menentukan sebaliknya, AWS DMS mengasumsikan `ColumnScale` adalah nol dan memotong detik fraksional.

Menggunakan CDC dengan Amazon S3 sebagai sumber untuk AWS DMS

Setelah AWS DMS melakukan beban data penuh, DMS secara opsional dapat mereplikasi perubahan data pada titik akhir target. Untuk melakukannya, Anda mengunggah file change data capture (file CDC) untuk Amazon S3 bucket Anda. AWS DMS membaca file CDC ini ketika Anda mengunggah file-file ini, dan kemudian menerapkan perubahan pada titik akhir target.

File CDC dinamakan sebagai berikut:

```
CDC00001.csv  
CDC00002.csv  
CDC00003.csv  
...
```

Note

Untuk mereplikasi file CDC di folder data perubahan unggahlah file-file ini secara berhasil dalam urutan leksikal (sekuensial). Sebagai contoh, unggah file CDC00002.csv sebelum file CDC00003.csv. Jika tidak, CDC00002.csv dilewati dan tidak direplikasi jika Anda memuat setelah CDC00003.csv. Tapi file CDC00004.csv berhasil bereplikasi jika dimuat setelah CDC00003.csv.

Untuk menunjukkan di mana AWS DMS dapat menemukan file, tentukan `cdcPath` parameter. Melanjutkan contoh sebelumnya, jika Anda mengatur `cdcPath` menjadi *changedata*, maka AWS DMS membaca file CDC di jalur berikut.

```
s3://mybucket/changedata
```

Jika Anda mengatur `cdcPath` kepada *changedata* dan `bucketFolder` kepada *myFolder*, maka AWS DMS membaca file CDC di jalur berikut.

```
s3://mybucket/myFolder/changedata
```

Catatan dalam file CDC diformat sebagai berikut:

- Operasi – perubahan yang akan dilakukan: INSERT atau I, UPDATE atau U, atau DELETE atau D. Nilai-nilai kata kunci dan karakter ini bersifat tidak peka huruf kapital.

Note

Dalam versi yang didukung AWS DMS, AWS DMS dapat mengidentifikasi operasi untuk melakukan setiap catatan beban dalam dua cara. AWS DMS dapat melakukan ini dari nilai kata kunci catatan (misalnya, INSERT) atau dari karakter awal kata kuncinya (misalnya, I). Dalam versi sebelumnya, AWS DMS mengakui operasi beban hanya dari nilai kata kunci penuh.

Dalam AWS DMS versi sebelumnya, nilai kata kunci penuh ditulis untuk log data CDC. Selain itu, versi sebelumnya menulis nilai operasi untuk setiap target S3 hanya menggunakan kata kunci awal.

Mengenali kedua format memungkinkan AWS DMS untuk menangani operasi terlepas dari bagaimana kolom operasi ditulis untuk membuat data sumber S3. Pendekatan ini mendukung penggunaan data target S3 sebagai sumber untuk migrasi berikutnya. Dengan pendekatan ini, Anda tidak perlu mengubah format nilai awal kata kunci yang muncul di kolom operasi sumber S3 berikutnya.

- Tabel nama - nama tabel sumber.
- Skema nama - nama skema sumber.
- Data - kolom satu atau lebih yang mewakili data yang akan diubah.

Berikut ini adalah contoh file CDC untuk tabel bernama 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
```

Prasyarat saat menggunakan Amazon S3 sebagai sumber untuk AWS DMS

Untuk menggunakan Amazon S3 sebagai sumber untuk AWS DMS, bucket S3 sumber Anda harus berada dalam Wilayah AWS sebagai instans replikasi DMS yang memigrasikan data Anda. Selain itu, akun AWS yang Anda gunakan untuk migrasi harus memiliki akses baca ke bucket sumber.

AWS Identity and Access Management (IAM) role yang diberikan kepada akun pengguna yang digunakan untuk membuat tugas migrasi harus memiliki serangkaian izin berikut.

```
{
  "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*"
    ]
  }
]
}

```

Yang AWS Identity and Access Management Peran (IAM) yang ditetapkan ke akun pengguna yang digunakan untuk membuat tugas migrasi harus memiliki set izin berikut jika versi diaktifkan pada bucket 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*"
      ]
    }
  ]
}

```

Batasan saat menggunakan Amazon S3 sebagai sumber AWS DMS

Batasan berikut berlaku saat menggunakan Amazon S3 sebagai sumber:

- Jangan aktifkan versi untuk S3. Jika Anda memerlukan versi S3, gunakan kebijakan siklus hidup untuk menghapus versi lama secara aktif. Jika tidak, Anda mungkin mengalami kegagalan koneksi pengujian endpoint karena S3 `list-object` panggilan batas waktu. Untuk membuat kebijakan siklus hidup untuk bucket S3, lihat [Mengelola siklus hidup penyimpanan](#). Untuk menghapus versi objek S3, lihat [Menghapus versi objek dari bucket versioning-enabled](#).
- Bucket S3 berkemampuan VPC (gateway VPC) didukung dalam versi 3.4.7 dan yang lebih tinggi.

Pengaturan titik akhir untuk Amazon S3 sebagai sumber AWS DMS

Anda dapat menggunakan pengaturan titik akhir untuk mengonfigurasi basis data sumber Amazon S3 yang serupa dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir sumber menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan `--s3-settings '{"EndpointSetting": "value", ...}'` sintaks JSON.

Tabel berikut menunjukkan pengaturan titik akhir yang dapat Anda gunakan dengan Amazon S3 sebagai sumber.

Pilihan	Deskripsi
BucketFolder	(Opsional) Nama folder dalam bucket S3. Jika atribut ini disediakan, file data sumber dan file CDC dibaca dari jalur <code>s3://myBucket/bucketFolder/schemaName/tableName</code> dan <code>s3://myBucket/bucketFolder/</code> masing-masing. Jika atribut ini tidak ditentukan, maka jalur yang digunakan adalah <code>schemaName/tableName/</code> . '{"BucketFolder": " <i>sourceData</i> "}'
BucketName	Nama bucket S3. '{"BucketName": " <i>myBucket</i> "}'
CdcPath	Lokasi file CDC. Atribut ini diperlukan jika tugas menangkap perubahan data; jika tidak, itu opsional. Jika CdcPath ada, maka AWS DMS

Pilihan	Deskripsi
	<p>membaca file CDC dari jalur ini dan mereplikasi perubahan data ke titik akhir target . Untuk informasi selengkapnya, lihat Menggunakan CDC dengan Amazon S3 sebagai sumber untuk AWS DMS.</p> <pre>'{"CdcPath": " <i>changeData</i> "'}</pre>
CsvDelimiter	<p>Pembatas digunakan untuk memisahkan kolom dalam file sumber. Default adalah koma. Contoh sebagai berikut.</p> <pre>'{"CsvDelimiter": ",","}'</pre>
CsvNullValue	<p>String yang ditentukan pengguna ituAWS DMSmemperlakukan sebagai nol saat membaca dari sumbernya. Default-nya adalah string kosong. Jika Anda tidak mengatur parameter ini,AWS DMSmemperlakukan string kosong sebagai nilai null. Jika Anda mengatur parameter ini ke string seperti "\ N",AWS DMSmemperlakukan string ini sebagai nilai null, dan memperlakukan string kosong sebagai nilai string kosong.</p>
CsvRowDelimiter	<p>Pembatas digunakan untuk memisahkan baris dalam file sumber. Default adalah newline (\n).</p> <pre>'{"CsvRowDelimiter": "\n"}'</pre>
IgnoreHeaderRows	<p>Ketika nilai ini diatur menjadi 1, AWS DMS mengabaikan header baris pertama dalam file .csv. Nilai 1 memungkinkan fitur tersebut, nilai 0 menonaktifkannya.</p> <p>Default adalah 0.</p> <pre>'{"IgnoreHeaderRows": 1}'</pre>

Pilihan	Deskripsi
Rfc4180	<p>Ketika nilai ini diatur ke <code>true</code>, setiap tanda kutip ganda terkemuka harus diikuti oleh tanda kutip ganda akhir. Format ini sesuai dengan RFC 4180. Ketika nilai ini diatur ke <code>false</code>, string literal disalin ke target sebagaimana adanya. Dalam kasus ini, pembatas (baris atau kolom) menandakan akhir bidang. Dengan demikian, Anda tidak dapat menggunakan pembatas sebagai bagian dari string, karena pembatas menandakan akhir nilai.</p> <p>Default adalah <code>true</code>.</p> <p>Nilai yang valid: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p> <p><code>'{"Rfc4180": false}'</code></p>

Jenis data sumber untuk Amazon S3

Migrasi data yang menggunakan Amazon S3 sebagai sumber untuk AWS DMS perlu memetakan data dari Amazon S3 ke jenis data AWS DMS. Untuk informasi selengkapnya, lihat [Mendefinisikan tabel eksternal untuk Amazon S3 sebagai sumber untuk AWS DMS](#).

Untuk informasi tentang cara melihat jenis data yang dipetakan dalam target, lihat bagian titik akhir target yang Anda gunakan.

Untuk informasi tambahan tentang jenis data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Jenis data AWS DMS berikut digunakan dengan Amazon S3 sebagai sumber:

- **BYTE** - Membutuhkan `ColumnLength`. Untuk informasi selengkapnya, lihat [Mendefinisikan tabel eksternal untuk Amazon S3 sebagai sumber untuk AWS DMS](#).
- **DATE**
- **WAKTU**
- **DATETIME** - Untuk informasi lebih lanjut dan contoh, lihat contoh tipe `DATETIME` di [Mendefinisikan tabel eksternal untuk Amazon S3 sebagai sumber untuk AWS DMS](#).
- **INT1**
- **INT2**

- INT4
- INT8
- NUMERIC - Membutuhkan `ColumnPrecision` dan `ColumnScale`. AWS DMS mendukung nilai maksimum berikut:
 - `ColumnPrecision`: 38
 - `ColumnScale`: 31

Untuk informasi lebih lanjut dan contoh, lihat contoh tipe NUMERIC di [Mendefinisikan tabel eksternal untuk Amazon S3 sebagai sumber untuk AWS DMS](#).

- REAL4
- REAL8
- STRING - Membutuhkan `ColumnLength`. Untuk informasi selengkapnya, lihat [Mendefinisikan tabel eksternal untuk Amazon S3 sebagai sumber untuk AWS DMS](#).
- UINT1
- UINT2
- UINT4
- UINT8
- BLOB
- CLOB
- BOOLEAN

Menggunakan IBM Db2 untuk Linux, Unix, Windows, dan database Amazon RDS (Db2 LUW) sebagai sumber untuk AWS DMS

Anda dapat memigrasikan data dari database IBM Db2 untuk Linux, Unix, Windows, dan Amazon RDS (Db2 LUW) ke database target yang didukung menggunakan (). AWS Database Migration Service AWS DMS

Untuk informasi tentang versi Db2 di Linux, Unix, Windows, dan RDS yang AWS DMS mendukung sebagai sumber, lihat. [Sumber untuk AWS DMS](#)

Anda dapat menggunakan Lapisan Soket Aman (SSL) untuk mengenkripsi sambungan antara titik akhir Db2 LUW dan instans replikasi. Untuk informasi lebih lanjut tentang menggunakan SSL dengan titik akhir Db2 LUW, lihat [Menggunakan SSL dengan AWS Database Migration Service](#).

Prasyarat saat menggunakan Db2 LUW sebagai sumber untuk AWS DMS

Prasyarat berikut diperlukan sebelum Anda dapat menggunakan basis data Db2 LUW sebagai sumber.

Untuk mengaktifkan replikasi yang sedang berlangsung, juga disebut sebagai change data capture (CDC), lakukan hal berikut:

- Atur database agar dapat dipulihkan, yang AWS DMS membutuhkan untuk menangkap perubahan. Database dapat dipulihkan jika salah satu atau kedua parameter konfigurasi database LOGARCHMETH1 dan LOGARCHMETH2 diatur ke. ON

Jika database Anda dapat dipulihkan, maka AWS DMS dapat mengakses ARCHIVE LOG Db2 jika diperlukan.

- Pastikan bahwa log transaksi DB2 tersedia, dengan periode retensi yang cukup untuk diproses oleh. AWS DMS
- DB2 memerlukan SYSADM atau DBADM otorisasi untuk mengekstrak catatan log transaksi. Berikan akun pengguna izin-izin berikut:
 - SYSADM atau DBADM
 - DATAACCESS

Note

Untuk tugas yang hanya memuat penuh, akun pengguna DMS memerlukan izin DATAACCESS.

- Bila menggunakan IBM DB2 untuk LUW versi 9.7 sebagai sumber, atur atribut sambungan tambahan (ECA), `CurrentLSN` sebagai berikut:

`CurrentLSN=LSN` di mana *LSN* menentukan log sequence number (LSN) di tempat Anda inginkan replikasi dimulai. Atau, `CurrentLSN=scan`.

Keterbatasan saat menggunakan Db2 LUW sebagai sumber untuk AWS DMS

AWS DMS tidak mendukung database berkerumun. Namun, Anda dapat menentukan Db2 LUW terpisah untuk setiap titik akhir klaster. Misalnya, Anda dapat membuat tugas migrasi Full Load dengan salah satu node di cluster, lalu membuat tugas terpisah dari setiap node.

AWS DMS tidak mendukung tipe BOOLEAN data dalam database Db2 LUW sumber Anda.

Saat menggunakan replikasi yang sedang berlangsung (CDC), keterbatasan berikut berlaku:

- Ketika tabel dengan beberapa partisi terpotong, jumlah peristiwa DDL yang ditampilkan di AWS DMS konsol sama dengan jumlah partisi. Hal ini karena Db2 LUW mencatat DDL terpisah untuk setiap partisi.
- Tindakan DDL berikut ini tidak didukung pada tabel yang dipartisi:
 - ALTER TABLE ADD PARTITION
 - ALTER TABLE DETACH PARTITION
 - ALTER TABLE ATTACH PARTITION
- AWS DMS tidak mendukung migrasi replikasi yang sedang berlangsung dari instance siaga pemulihan bencana ketersediaan tinggi (HADR) DB2. Keadaan siaga tidak dapat diakses.
- Jenis data DECFLOAT tidak didukung. Akibatnya, perubahan pada kolom DECFLOAT diabaikan selama replikasi sedang berlangsung.
- Pernyataan RENAME COLUMN tidak didukung.
- Saat melakukan pembaruan pada tabel Multi-Dimensional Clustering (MDC), setiap pembaruan ditampilkan di AWS DMS konsol sebagai INSERT + DELETE.
- Ketika pengaturan tugas Sertakan kolom LOB dalam replikasi tidak diaktifkan, setiap tabel yang memiliki kolom LOB ditangguhkan selama replikasi sedang berlangsung.
- Untuk Db2 LUW versi 10.5 dan yang lebih tinggi, kolom string panjang variabel dengan data yang disimpan diabaikan. out-of-row Batasan ini hanya berlaku untuk tabel yang dibuat dengan ukuran baris yang diperluas untuk kolom dengan tipe data seperti VARCHAR dan VARGRAPHIC. Untuk mengatasi batasan ini, pindahkan tabel ke ruang tabel dengan ukuran halaman yang lebih tinggi. Untuk informasi lebih lanjut, lihat [Apa yang dapat saya lakukan jika saya ingin mengubah ukuran halaman ruang tabel DB2](#).
- Untuk replikasi yang sedang berlangsung, DMS tidak mendukung migrasi data yang dimuat di tingkat halaman oleh utilitas DB2 LOAD. Sebaliknya, gunakan utilitas IMPORT yang menggunakan sisipan SQL. Untuk informasi selengkapnya, lihat [perbedaan antara utilitas impor dan beban](#).
- Saat tugas replikasi sedang berjalan, DMS menangkap CREATE TABLE DDL hanya jika tabel dibuat dengan atribut DATA CAPTURE CHANGE.
- DMS memiliki batasan berikut saat menggunakan Db2 Database Partition Feature (DPF):
 - DMS tidak dapat mengoordinasikan transaksi di seluruh node Db2 di lingkungan DPF. Ini karena kendala dalam antarmuka API IBM DB2READLOG. Dalam DPF, transaksi dapat menjangkau

beberapa node Db2, tergantung pada bagaimana DB2 mempartisi data. Akibatnya, solusi DMS Anda harus menangkap transaksi dari setiap node Db2 secara independen.

- DMS dapat menangkap transaksi lokal dari setiap node Db2 di cluster DPF dengan menyetel `connectNode` ke 1 beberapa titik akhir sumber DMS. Konfigurasi ini sesuai dengan nomor node logis yang didefinisikan dalam file konfigurasi server DB2. `db2nodes.cfg`
- Transaksi lokal pada node Db2 individu dapat menjadi bagian dari transaksi global yang lebih besar. DMS menerapkan setiap transaksi lokal secara independen pada target, tanpa koordinasi dengan transaksi pada node Db2 lainnya. Pemrosesan independen ini dapat menyebabkan komplikasi, terutama ketika baris dipindahkan antar partisi.
- Ketika DMS mereplikasi dari beberapa node Db2, tidak ada jaminan urutan operasi yang benar pada target, karena DMS menerapkan operasi secara independen untuk setiap node Db2. Anda harus memastikan bahwa menangkap transaksi lokal secara independen dari setiap node Db2 berfungsi untuk kasus penggunaan spesifik Anda.
- Saat bermigrasi dari lingkungan DPF, sebaiknya jalankan tugas Full Load terlebih dahulu tanpa peristiwa yang di-cache, lalu menjalankan tugas khusus CDC. Sebaiknya jalankan satu tugas per node Db2, mulai dari stempel waktu mulai Full Load atau LRI (pengenal catatan log) yang Anda atur menggunakan pengaturan titik akhir. `StartFromContext` Untuk informasi tentang menentukan titik awal replikasi Anda, lihat [Menemukan nilai LSN atau LRI untuk permulaan replikasi di](#) dokumentasi IBM Support.
- Untuk replikasi berkelanjutan (CDC), jika Anda berencana untuk memulai replikasi dari stempel waktu tertentu, Anda harus menyetel atribut `StartFromContext` koneksi ke stempel waktu yang diperlukan.
- Saat ini, DMS tidak mendukung Fitur Db2 PureScale, ekstensi DB2 LUW yang dapat Anda gunakan untuk menskalakan solusi database Anda.
- AWS DMS tidak mendukung CDC saat menggunakan Db2 untuk Amazon RDS sebagai sumber.

Pengaturan titik akhir saat menggunakan Db2 LUW sebagai sumber untuk AWS DMS

Anda dapat menggunakan pengaturan titik akhir untuk mengonfigurasi basis data sumber Db2 LUW Anda mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir sumber menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--ibm-db2-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan endpoint yang dapat Anda gunakan dengan Db2 LUW sebagai sumber.

Nama	Deskripsi
<code>CurrentLSN</code>	Untuk replikasi yang sedang berlangsung (CDC), gunakan <code>CurrentLSN</code> untuk menentukan log sequence number (LSN) di tempat Anda ingin replikasi dimulai.
<code>MaxKBytesPerRead</code>	Jumlah maksimum byte per baca, sebagai nilai <code>NUMBER</code> . Default adalah 64 KB.
<code>SetDataCaptureChanges</code>	Memungkinkan replikasi yang sedang berlangsung (CDC) sebagai nilai <code>BOOLEAN</code> . Default adalah benar.
<code>StartFromContext</code>	<p>Untuk replikasi berkelanjutan (CDC), gunakan <code>StartFromContext</code> untuk menentukan batas bawah log dari mana memulai replikasi. <code>StartFromContext</code> menerima berbagai bentuk nilai. Nilai yang valid meliputi:</p> <ul style="list-style-type: none"> <code>timestamp</code> (UTC). Sebagai contoh: <div data-bbox="721 1157 1507 1276" data-label="Code-Block"> <pre>'{"StartFromContext": "timestamp:2021-09-21T13:00:00"}'</pre> </div> <code>NOW</code> <p>Untuk IBM DB2 LUW versi 10.5 dan lebih tinggi, <code>SEKARANG</code> dikombinasikan dengan <code>currentLSN: scan</code>, memulai tugas dari LSO terbaru. Sebagai contoh: <div data-bbox="721 1591 1507 1711" data-label="Code-Block"> <pre>'{"CurrentLSN": "scan", "StartFromContext": "NOW"}'</pre> </div> </p> LRI tertentu. Sebagai contoh:

Nama	Deskripsi
	<pre data-bbox="721 212 1507 327">'{"StartFromContext": "0100000000000022C C0000000000004FB13"}'</pre> <p data-bbox="688 396 1503 527">Untuk menentukan rentang LRI/LSN dari file log, jalankan db2f1sn perintah seperti yang ditunjukkan pada contoh berikut.</p> <pre data-bbox="695 562 1507 646">db2f1sn -db <i>SAMPLE</i> -lri range 2</pre> <p data-bbox="688 680 1419 716">Output dari contoh itu mirip dengan yang berikut ini.</p> <pre data-bbox="695 751 1507 989">S0000002.LOG: has LRI range 00000000000000100 0000000000225400000000004F9A6 to 000000000000001000000000022CC0000000004 FB13</pre> <p data-bbox="688 1031 1495 1115">Dalam output itu, file log adalah S0000002.LOG dan nilai StartFromContextLRI adalah 34 byte di akhir rentang.</p> <pre data-bbox="695 1150 1507 1234">010000000000022CC00000000004FB13</pre>

Jenis data sumber untuk IBM Db2 LUW

Migrasi data yang menggunakan Db2 LUW sebagai sumber untuk AWS DMS mendukung sebagian besar tipe data Db2 LUW. Tabel berikut menunjukkan tipe data sumber Db2 LUW yang didukung saat menggunakan AWS DMS dan pemetaan default dari tipe data. AWS DMS Untuk informasi lebih lanjut tentang jenis data Db2 LUW, lihat [Dokumentasi Db2 LUW](#).

Untuk informasi tentang cara melihat jenis data yang dipetakan dalam target, lihat bagian titik akhir target yang Anda gunakan.

Untuk informasi tambahan tentang tipe AWS DMS data, lihat [Tipe data untuk AWS Database Migration Service](#).

Jenis data Db2 LUW	AWS DMS tipe data
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
DECIMAL (p,s)	NUMERIC (p,s)
FLOAT	REAL8
DOUBLE	REAL8
REAL	REAL4
DECFLOAT (p)	Jika presisi adalah 16, maka REAL8; jika presisi adalah 34, maka STRING
GRAPHIC (n)	WSTRING, untuk string grafis yang panjangnya tetap dari karakter byte ganda dengan panjang lebih besar dari 0 dan kurang dari atau sama dengan 127
VARGRAPHIC (n)	WSTRING, untuk string grafis yang panjangnya beragam dengan panjang lebih besar dari 0 dan kurang dari atau sama dengan 16.352 karakter byte ganda
LONG VARGRAPHIC (n)	CLOB, untuk string grafis yang panjangnya beragam dengan panjang lebih besar dari 0 dan kurang dari atau sama dengan 16.352 karakter byte ganda
CHARACTER (n)	STRING, untuk string karakter byte ganda yang panjangnya tetap dengan panjang lebih besar dari 0 dan kurang dari atau sama dengan 255
VARCHAR (n)	STRING, untuk string karakter byte ganda yang panjangnya beragam dengan panjang lebih

Jenis data Db2 LUW	AWS DMS tipe data
	besar dari 0 dan kurang dari atau sama dengan 32.704
LONG VARCHAR (n)	CLOB, untuk string karakter byte ganda yang panjangnya beragam dengan panjang lebih besar dari 0 dan kurang dari atau sama dengan 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 Panjang maksimum adalah 2.147.483.647 byte
CLOB (n)	CLOB Panjang maksimum adalah 2.147.483.647 byte
DBCLOB (n)	CLOB Panjang maksimum adalah 1.073.741.824 karakter byte ganda
XML	CLOB

Menggunakan IBM Db2 untuk database z/OS sebagai sumber untuk AWS DMS

Anda dapat memigrasi data dari database IBM untuk z/OS ke database target yang didukung menggunakan AWS Database Migration Service (). AWS DMS

Untuk informasi tentang versi Db2 untuk z/OS yang AWS DMS mendukung sebagai sumber, lihat. [Sumber untuk AWS DMS](#)

Prasyarat saat menggunakan Db2 untuk z/OS sebagai sumber untuk AWS DMS

Untuk menggunakan basis data IBM Db2 untuk basis data z/OS sebagai sumber di AWS DMS, berikan hak istimewa berikut untuk Db2 untuk pengguna z/OS yang ditentukan dalam pengaturan sambungan titik akhir sumber.

```
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
```

Juga memberikan SELECT ON tabel *user defined* sumber.

AWS DMS IBM Db2 untuk titik akhir sumber z/OS bergantung pada IBM Data Server Driver untuk ODBC untuk mengakses data. Server database harus memiliki lisensi IBM ODBC Connect yang valid untuk DMS untuk terhubung ke titik akhir ini.

Keterbatasan saat menggunakan Db2 untuk z/OS sebagai sumber untuk AWS DMS

Keterbatasan berikut berlaku saat menggunakan basis data IBM Db2 untuk basis data Z/OS sebagai sumber untuk: AWS DMS

- Hanya tugas replikasi Beban Penuh yang didukung. Change data capture (CDC) tidak didukung.
- Beban paralel tidak didukung.
- Validasi data tampilan tidak didukung.
- Nama skema, tabel, dan kolom harus ditentukan dalam kasus UPPER dalam pemetaan tabel untuk transformasi tingkat kolom/tabel dan filter pemilihan tingkat baris.

Jenis data sumber untuk IBM Db2 untuk z/OS

migrasi data yang menggunakan Db2 untuk z/OS sebagai sumber untuk AWS DMS dukungan yang paling Db2 untuk tipe data z/OS. Tabel berikut menunjukkan Db2 untuk jenis data sumber z/OS yang didukung saat menggunakan AWS DMS, dan pemetaan default dari jenis data. AWS DMS

Untuk informasi lebih lanjut tentang Db2 untuk jenis data z/OS, lihat Dokumentasi [IBM Db2](#) untuk z/OS.

Untuk informasi tentang cara melihat jenis data yang dipetakan dalam target, lihat bagian titik akhir target yang Anda gunakan.

Untuk informasi tambahan tentang jenis data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Db2 untuk tipe data z/OS	Jenis data AWS DMS
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
DECIMAL (p,s)	NUMERIC (p,s) Jika titik desimal diatur ke koma (,) dalam konfigurasi DB2, mengkonfigurasi Replicate untuk mendukung pengaturan DB2.
MENGAMBANG	REAL8
DOUBLE	REAL8
REAL	REAL4
DECFLOAT (p)	Jika presisi adalah 16, maka REAL8; jika presisi adalah 34, maka STRING
GRAPHIC (n)	Jika $n \geq 127$ kemudian WSTRING, untuk string grafis yang panjangnya tetap dari karakter byte ganda dengan panjang lebih

Db2 untuk tipe data z/OS	Jenis data AWS DMS
	besar dari 0 dan kurang dari atau sama dengan 127
VARGRAPHIC (n)	WSTRING, untuk string grafis yang panjangnya beragam dengan panjang lebih besar dari 0 dan kurang dari atau sama dengan 16.352 karakter byte ganda
LONG VARGRAPHIC (n)	CLOB, untuk string grafis yang panjangnya beragam dengan panjang lebih besar dari 0 dan kurang dari atau sama dengan 16.352 karakter byte ganda
CHARACTER (n)	STRING, untuk string karakter byte ganda yang panjangnya tetap dengan panjang lebih besar dari 0 dan kurang dari atau sama dengan 255
VARCHAR (n)	STRING, untuk string karakter byte ganda yang panjangnya beragam dengan panjang lebih besar dari 0 dan kurang dari atau sama dengan 32.704
LONG VARCHAR (n)	CLOB, untuk string karakter byte ganda yang panjangnya beragam dengan panjang lebih besar dari 0 dan kurang dari atau sama dengan 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

Db2 untuk tipe data z/OS	Jenis data AWS DMS
BLOB (n)	BLOB Panjang maksimum adalah 2.147.483.647 byte
CLOB (n)	CLOB Panjang maksimum adalah 2.147.483.647 byte
DBCLOB (n)	CLOB Panjang maksimum adalah 1.073.741.824 karakter byte ganda
XML	CLOB
BINER	BYTES
VARBINARY	BYTES
ROWID	BYTE. Untuk informasi lebih lanjut tentang bekerja dengan ROWID, lihat berikut.
TIMESTAMP WITH TIME ZONE	Tidak didukung.

Kolom ROWID dimigrasi secara default ketika mode persiapan tabel target untuk tugas diatur ke `DROP_AND_CREATE` (default). Validasi data mengabaikan kolom ini karena baris tidak berarti di luar database dan tabel tertentu. Untuk menonaktifkan migrasi kolom ini, Anda dapat melakukan salah satu langkah persiapan berikut:

- Precreate tabel target tanpa kolom ini. Kemudian, atur mode persiapan tabel target tugas ke `DO_NOTHING` atau `TRUNCATE_BEFORE_LOAD`. Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk membuat tabel target tanpa kolom.
- Tambahkan aturan pemetaan tabel ke tugas yang menyaring kolom ini sehingga diabaikan. Untuk informasi selengkapnya, lihat [Aturan dan tindakan transformasi](#).

Kolasi EBCDIC di PostgreSQL untuk layanan Modernisasi Mainframe AWS

AWS Program Modernisasi Mainframe membantu Anda memodernisasi aplikasi mainframe Anda ke lingkungan runtime yang dikelola. AWS Ini menyediakan alat dan sumber daya yang membantu Anda merencanakan dan mengimplementasikan proyek migrasi dan modernisasi Anda. Untuk informasi selengkapnya tentang modernisasi dan migrasi mainframe, lihat Modernisasi [Mainframe](#) dengan AWS

Beberapa IBM Db2 untuk set data z/OS dikodekan dalam set karakter Extended Binary Coded Decimal Interchange (EBCDIC). Ini adalah set karakter yang dikembangkan sebelum ASCII (American Standard Code for Information Interchange) menjadi umum digunakan. Halaman kode memetakan setiap karakter teks ke karakter dalam satu set karakter. Halaman kode tradisional berisi informasi pemetaan antara titik kode dan ID karakter. Sebuah ID karakter adalah 8-byte karakter string data. Titik kode adalah bilangan biner 8-bit yang mewakili karakter. Poin kode biasanya ditampilkan sebagai representasi heksadesimal dari nilai biner mereka.

Jika saat ini Anda menggunakan Micro Focus atau BluAge komponen layanan Modernisasi Mainframe, Anda harus memberi tahu AWS DMS untuk menggeser (menerjemahkan) titik kode tertentu. Anda dapat menggunakan pengaturan AWS DMS tugas untuk melakukan shift. Contoh berikut menunjukkan bagaimana menggunakan AWS DMS `CharacterSetSettings` operasi untuk memetakan pergeseran dalam pengaturan tugas 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"}
  ]
}
```

Beberapa kolasi EBCDIC sudah ada untuk PostgreSQL yang memahami pergeseran yang diperlukan. Beberapa halaman kode yang berbeda didukung. Bagian berikut memberikan contoh

JSON tentang apa yang harus Anda geser untuk semua halaman kode yang didukung. Anda hanya dapat JSON copy-and-past yang diperlukan yang Anda butuhkan dalam tugas DMS Anda.

Kolasi EBCDIC spesifik Fokus Mikro

Untuk Micro Focus, geser subset karakter sesuai kebutuhan untuk kolasi berikut.

```
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 Data Fokus Mikro bergeser per pemeriksaan:

en_us_cp1140m

Kode Shift:

```
0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178
```

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{ "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

Kode Shift:

0000	0180
00B8	0160
00BC	0161
00BD	017D
00BE	017E
00A8	0152
00B4	0153
00A6	0178

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{ "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

Kode Shift:

0000	0180
00A6	0160
00B8	0161
00BC	017D
00BD	017E
00BE	0152
00A8	0153
00B4	0178

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{ "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

Kode Shift:

0000	0180
00B8	0160
00BC	0161
00BD	017D
00BE	017E
00A8	0152
00B4	0153
00A6	0178

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{ "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" }
```

cp1144m

Kode Shift:

```

0000    0180
00B8    0160
00BC    0161
00BD    017D
00BE    017E
00A8    0152
00B4    0153
00A6    0178

```

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```

{"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

Kode Shift:

```

0000    0180
00A6    0160
00B8    0161
00A8    017D
00BC    017E
00BD    0152
00BE    0153
00B4    0178

```

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```

{"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

Kode Shift:

```
0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178
```

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{"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

Kode Shift:

```
0000    0180
00B8    0160
00A8    0161
00BC    017D
00BD    017E
```

```

00BE    0152
00B4    0153
00A6    0178

```

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```

{"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

Kode Shift:

```

0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178

```

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```

{"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"}

```

BluAgekolasi EBCDIC tertentu

Untuk BluAge, geser semua nilai rendah dan nilai tinggi berikut sesuai kebutuhan. Kolasi ini seharusnya hanya digunakan untuk mendukung layanan Migrasi Mainframe. 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 BluAgePergeseran Data:

DA-DK-CP277B dan DA-DK-CP1142B

Kode Shift:

```
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

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{ "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 dan de-de-1141b

Kode Shift:

```
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
```

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{ "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"}
```

ID-GB-285b dan id-GB-1146b

Kode Shift:

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
```

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{"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"}
```

id-us-037b dan id-us-1140b

Kode Shift:

```
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

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{"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 dan ES-ES-1145B

Kode Shift:

```
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

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{ "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 dan fi-fi-1143b

Kode Shift:

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

```

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```

{"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 dan FR-FR-1147B

Kode Shift:

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

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{"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"}
```

itu-itu-280b dan itu-itu-1144b

Kode Shift:

```
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

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{ "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-menjadi-500b dan NL-menjadi-1148b

Kode Shift:

```
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

Pemetaan masukan yang sesuai untuk AWS DMS tugas:

```
{ "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"}
```

Target migrasi data

AWS Database Migration Service (AWS DMS) dapat menggunakan banyak basis data yang paling populer sebagai target replikasi data. Target dapat berada di instans Amazon Elastic Compute Cloud (Amazon EC2), instans Amazon Relational Database Service (Amazon RDS), atau basis data on-premise.

Untuk daftar target valid selengkapnya, lihat [Target untuk AWS DMS](#).

Note

AWS DMS tidak support migrasi di wilayah AWS untuk jenis titik akhir target berikut:

- Amazon DynamoDB
- OpenSearch Layanan Amazon
- Amazon Kinesis Data Streams

Topik

- [Menggunakan basis data Oracle sebagai target untuk AWS Database Migration Service](#)
- [Menggunakan basis data Microsoft SQL Server sebagai target untuk AWS Database Migration Service](#)
- [Menggunakan basis data PostgreSQL sebagai target untuk AWS Database Migration Service](#)
- [Menggunakan basis data yang kompatibel dengan MySQL sebagai target untuk AWS Database Migration Service](#)
- [Menggunakan basis data Amazon Redshift sebagai target untuk AWS Database Migration Service](#)
- [Menggunakan basis data SAP ASE sebagai target untuk AWS Database Migration Service](#)
- [Menggunakan Amazon S3 sebagai target untuk AWS Database Migration Service](#)
- [Menggunakan basis data Amazon DynamoDB sebagai target untuk AWS Database Migration Service](#)
- [Menggunakan Amazon Kinesis Data Streams sebagai target AWS Database Migration Service](#)
- [Menggunakan Apache Kafka sebagai target AWS Database Migration Service](#)
- [Menggunakan AmazonOpenSearchService cluster sebagai targetAWS Database Migration Service](#)
- [Menggunakan Amazon DocumentDB sebagai target untuk Database Migration Service AWS](#)
- [Menggunakan Amazon Neptune sebagai target untuk AWS Database Migration Service](#)
- [Menggunakan Redis sebagai target AWS Database Migration Service](#)
- [Menggunakan Babelfish sebagai target untuk AWS Database Migration Service](#)
- [Menggunakan Amazon Timestream sebagai target AWS Database Migration Service](#)
- [Menggunakan Amazon RDS untuk Db2 dan IBM Db2 LUW sebagai target AWS DMS](#)

Menggunakan basis data Oracle sebagai target untuk AWS Database Migration Service

Anda dapat memigrasikan data ke target database Oracle menggunakan AWS DMS, baik dari database Oracle lain atau dari salah satu database lain yang didukung. Anda dapat menggunakan

Lapisan Soket Aman (SSL) untuk mengenkripsi koneksi antara titik akhir Oracle dan instans replikasi. Untuk informasi lebih lanjut tentang menggunakan SSL dengan endpoint Oracle, lihat [Menggunakan SSL dengan AWS Database Migration Service](#). AWS DMS juga mendukung penggunaan enkripsi data transparan Oracle (TDE) untuk mengenkripsi data saat istirahat di database target karena Oracle TDE tidak memerlukan kunci enkripsi atau kata sandi untuk menulis ke database.

Untuk informasi tentang versi Oracle yang AWS DMS mendukung sebagai target, lihat [Target untuk AWS DMS](#).

Ketika Anda menggunakan Oracle sebagai target, kami berasumsi bahwa data tersebut akan dimigrasi ke dalam skema atau pengguna yang digunakan sebagai koneksi target. Jika Anda ingin melakukan migrasi data ke skema yang berbeda, lakukan dengan menggunakan transformasi skema. Contohnya, anggaplah titik akhir target Anda terhubung ke pengguna RDSMASTER dan Anda ingin melakukan migrasi dari pengguna PERFDATA1 ke PERFDATA2. Dalam situasi ini buatlah transformasi seperti contoh berikut.

```
{
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "rule-action": "rename",
  "rule-target": "schema",
  "object-locator": {
    "schema-name": "PERFDATA1"
  },
  "value": "PERFDATA2"
}
```

Saat menggunakan Oracle sebagai target, AWS DMS memigrasikan semua tabel dan indeks ke tabel default dan indeks ruang tabel di target. Jika Anda ingin melakukan migrasi tabel dan indeks ke tabel dan tablespace indeks yang berbeda, lakukan dengan menggunakan transformasi tablespace. Contohnya, anggaplah Anda memiliki seperangkat tabel di skema INVENTORY yang ditetapkan untuk beberapa tablespace dalam sumber Oracle. Untuk migrasi, Anda ingin menetapkan semua tabel ini untuk satu tablespace INVENTORYSPACE di dalam target. Dalam situasi ini, buat transformasi seperti berikut.

```
{
  "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"
}
```

Untuk informasi lebih lanjut tentang transformasi, lihat [Menentukan pemilihan tabel dan transformasi aturan menggunakan JSON](#).

Jika Oracle berperan sebagai sumber dan target, Anda dapat mempertahankan tabel atau penugasan tablespace indeks yang ada dengan mengatur atribut koneksi tambahan sumber Oracle, `enableHomogenousTablespace=true`. Untuk informasi lebih lanjut, lihat [Pengaturan titik akhir saat menggunakan Oracle sebagai sumber AWS DMS](#)

Untuk detail tambahan tentang bekerja dengan database Oracle sebagai target AWS DMS, lihat bagian berikut:

Topik

- [Keterbatasan pada Oracle sebagai target AWS Database Migration Service](#)
- [Hak akun pengguna diperlukan untuk menggunakan Oracle sebagai target](#)
- [Mengkonfigurasi database Oracle sebagai target AWS Database Migration Service](#)
- [Pengaturan titik akhir saat menggunakan Oracle sebagai target AWS DMS](#)
- [Target tipe data untuk Oracle](#)

Keterbatasan pada Oracle sebagai target AWS Database Migration Service

Keterbatasan saat menggunakan Oracle sebagai target untuk migrasi data meliputi hal berikut:

- AWS DMS tidak membuat skema pada database Oracle target. Anda harus membuat skema yang Anda inginkan di target basis data Oracle. Nama skema harus sudah ada untuk target Oracle. Tabel dari skema sumber diimpor ke pengguna atau skema, yang AWS DMS digunakan untuk terhubung ke instance target. Untuk memigrasikan beberapa skema, Anda dapat membuat

beberapa tugas replikasi. Anda juga dapat memigrasikan data ke skema yang berbeda pada target. Untuk melakukan ini, Anda perlu menggunakan aturan transformasi skema pada pemetaan AWS DMS tabel.

- AWS DMS tidak mendukung Use `direct path full load` opsi untuk tabel dengan `INDEXTYPE CONTEXT`. Sebagai solusi, Anda dapat menggunakan beban array.
- Dengan pilihan penerapan batch yang dioptimalkan, memuat ke tabel net changes akan melalui jalur langsung yang tidak mendukung tipe XML. Sebagai solusi, Anda dapat menggunakan mode penerapan transaksional.
- String kosong yang dimigrasi dari basis data sumber dapat diperlakukan berbeda oleh target Oracle (contoh: dikonversi ke string satu ruang). Hal ini dapat mengakibatkan AWS DMS validasi melaporkan ketidakcocokan.
- Anda dapat mengekspresikan jumlah kolom per tabel yang didukung dalam mode penerapan yang dioptimalkan oleh Batch, menggunakan rumus berikut:

```
2 * columns_in_original_table + columns_in_primary_key <= 999
```

Misalnya, jika tabel asli memiliki 25 kolom dan Kunci Primernya terdiri dari 5 kolom, maka jumlah kolom adalah 55. Jika tabel melebihi jumlah kolom yang didukung, maka semua perubahan diterapkan dalam one-by-one mode.

- AWS DMS tidak mendukung Autonomous DB di Oracle Cloud Infrastructure (OCI).

Hak akun pengguna diperlukan untuk menggunakan Oracle sebagai target

Untuk menggunakan target Oracle dalam AWS Database Migration Service tugas, berikan hak istimewa berikut dalam database Oracle. Anda memberikan itu pada akun pengguna yang telah ditentukan dalam definisi basis data Oracle untuk 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
- HAPUS TABEL APA PUN

Untuk persyaratan berikut, berikan tambahan hak istimewa tambahan:

- Untuk menggunakan daftar tabel tertentu, berikan SELECT pada setiap tabel yang direplikasi dan ALTER pada setiap tabel yang direplikasi.
- Untuk mengizinkan pengguna membuat tabel di tablespace default, berikan hak GRANT UNLIMITED TABLESPACE.
- Untuk masuk, berikan hak CREATE SESSION.
- Jika Anda menggunakan jalur langsung (yang merupakan default untuk beban penuh), GRANT LOCK ANY TABLE to *dms_user*;
- Jika skema berbeda saat menggunakan mode persiapan tabel "DROP and CREATE",. GRANT CREATE ANY INDEX to *dms_user*;

- Untuk beberapa skenario beban penuh, Anda dapat memilih pilihan “DROP and CREATE table” atau “TRUNCATE before loading” ketika skema target tabel berbeda dari target pengguna DMS. Dalam kasus ini, berikan hak DROP ANY TABLE.
- Untuk menyimpan perubahan dalam tabel perubahan atau tabel audit yang skema table targetnya berbeda dengan pengguna DMS, berikan CREATE ANY TABLE dan CREATE ANY INDEX.

Hak istimewa baca yang diperlukan untuk AWS Database Migration Service basis data target

Akun AWS DMS pengguna harus diberikan izin baca untuk tabel DBA berikut:

- 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
- PILIH pada SYS.DBA_REGISTRY

Jika salah satu hak yang diperlukan tidak dapat diberikan kepada V\$xxx, maka berikan hak tersebut pada V_\$xxx.

Penilaian Premigrasi

Untuk menggunakan penilaian premi yang tercantum [Penilaian Oracle](#) dengan Oracle sebagai Target, Anda harus menambahkan izin berikut ke pengguna `dms_user` database pada database target:

```
GRANT SELECT ON V_$INSTANCE TO dms_user;
```

Mengkonfigurasi database Oracle sebagai target AWS Database Migration Service

Sebelum menggunakan database Oracle sebagai target migrasi data, Anda harus memberikan akun pengguna Oracle. AWS DMS Akun pengguna harus memiliki hak baca/tulis pada basis data Oracle,

sebagaimana disebutkan dalam [Hak akun pengguna diperlukan untuk menggunakan Oracle sebagai target](#).

Pengaturan titik akhir saat menggunakan Oracle sebagai target AWS DMS

Anda dapat menggunakan pengaturan endpoint untuk mengkonfigurasi database target Oracle Anda mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir target menggunakan AWS DMS konsol, atau dengan menggunakan create-endpoint perintah di [AWS CLI](#), dengan sintaks `--oracle-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan endpoint yang dapat Anda gunakan dengan Oracle sebagai target.

Nama	Penjelasan
EscapeCharacter	<p>Atur atribut ini ke karakter escape. Karakter escape ini memungkinkan Anda untuk membuat karakter wildcard tunggal berperilaku sebagai karakter normal dalam ekspresi pemetaan tabel. Untuk informasi selengkapnya, lihat Wildcard dalam pemetaan tabel.</p> <p>Nilai default: Null</p> <p>Nilai yang valid: Karakter apa pun selain karakter wildcard</p> <p>Contoh: <code>--oracle-settings '{"EscapeCharacter": "#"}'</code></p>
UseDirectPathFullLoad	<p>Ketika diatur ke Y, AWS DMS menggunakan jalur langsung beban penuh. Tentukan nilai ini untuk mengaktifkan protokol jalur langsung di Oracle Call Interface (OCI). Protokol OCI ini memungkinkan pemuatan massal tabel target Oracle selama beban penuh.</p> <p>Nilai default: true</p> <p>Nilai valid: true/false</p>

Nama	Penjelasan
DirectPathParallelLoad	<p data-bbox="686 212 1325 296">Contoh: <code>--oracle-settings '{"UseDirectPathFullLoad": false}'</code></p> <p data-bbox="686 342 1490 615">Ketika diatur ke <code>true</code>, atribut ini menentukan beban paralel saat <code>UseDirectPathFullLoad</code> diatur ke <code>Y</code>. Atribut ini juga hanya berlaku ketika Anda menggunakan fitur AWS DMS parallel load. Untuk informasi lebih lanjut, lihat deskripsi operasi <code>parallel-load</code> di Tabel dan koleksi pengaturan aturan dan operasi.</p> <p data-bbox="686 661 1498 978">Sebuah kekurangan pada penentuan pengaturan beban paralel ini adalah bahwa tabel target tidak dapat memiliki batasan atau indeks. Untuk informasi lebih lanjut tentang keterbatasan ini, lihat Mengaktifkan Pembatasan Setelah Beban Jalur Langsung Paralel. Jika batasan atau indeks diaktifkan, pengaturan atribut ini ke <code>true</code> tidak berpengaruh.</p> <p data-bbox="686 1024 967 1058">Nilai default: <code>false</code></p> <p data-bbox="686 1104 1023 1138">Nilai valid: <code>true/false</code></p> <p data-bbox="686 1184 1325 1268">Contoh: <code>--oracle-settings '{"DirectPathParallelLoad": true}'</code></p>

Nama	Penjelasan
DirectPathNoLog	<p>Ketika diatur ke <code>true</code>, atribut ini membantu meningkatkan kecepatan melakukan pada basis data target Oracle dengan menulis langsung ke tabel dan tidak meninggalkan jejak ke basis data log. Untuk informasi lebih lanjut, lihat Direct-Load INSERT. Atribut ini juga hanya berlaku ketika Anda mengatur <code>UseDirectPathFullLoad</code> ke <code>Y</code>.</p> <p>Nilai default: <code>false</code></p> <p>Nilai valid: <code>true/false</code></p> <p>Contoh: <code>--oracle-settings '{"DirectPathNoLog": true}'</code></p>
CharLengthSemantics	<p>Menentukan apakah panjang kolom karakter dalam bytes atau character. Untuk menunjukkan bahwa panjang kolom karakter dalam character, atur atribut ini menjadi <code>CHAR</code>. Jika tidak, maka panjang kolom karakter adalah bytes.</p> <p>Nilai default: Tidak diatur ke <code>CHAR</code></p> <p>Nilai valid: <code>CHAR</code></p> <p>Contoh: <code>--oracle-settings '{"CharLengthSemantics": "CHAR"}'</code></p>

Nama	Penjelasan
AlwaysReplaceEmptyString	<p>AWS DMS menambahkan ruang ekstra untuk mereplikasi string kosong saat bermigrasi ke target Oracle. Secara umum, Oracle tidak memiliki notasi untuk string kosong. Saat Anda memasukkan string kosong pada varchar2, Anda memuat string kosong sebagai NULL. Jika Anda ingin menyisipkan data sebagai NULL di Oracle, atur atribut ini ke FALSE.</p> <p>Nilai default: true</p> <p>Nilai valid: true/false</p> <p>Contoh: <code>--oracle-settings '{"AlwaysReplaceEmptyString": false}'</code></p>

Target tipe data untuk Oracle

Database Oracle target yang digunakan dengan AWS DMS mendukung sebagian besar tipe data Oracle. Tabel berikut menunjukkan tipe data target Oracle yang didukung saat menggunakan AWS DMS dan pemetaan default dari tipe AWS DMS data. Untuk informasi lebih lanjut tentang cara melihat tipe data yang dipetakan dari sumber, lihat di bagian sumber yang Anda gunakan.

AWS DMS tipe data	Tipe data Oracle
BOOLEAN	NUMBER (1)
BYTES	RAW (length)
DATE	DATETIME
TIME	TIMESTAMP (0)
DATETIME	TIMESTAMP (scale)
INT1	NUMBER (3)
INT2	NUMBER (5)

AWS DMS tipe data	Tipe data Oracle
INT4	NUMBER (10)
INT8	NUMBER (19)
NUMERIC	NUMBER (p,s)
REAL4	FLOAT
REAL8	FLOAT
STRING	<p>Dengan indikasi tanggal: DATE</p> <p>Dengan indikasi waktu: TIMESTAMP</p> <p>Dengan indikasi timestamp: TIMESTAMP</p> <p>Dengan indikasi timestamp_with_timezone: TIMESTAMP WITH TIMEZONE</p> <p>Dengan indikasi timestamp_with_local_timezone: TIMESTAMP WITH LOCAL TIMEZONE. Dengan indikasi interval_year_to_month: INTERVAL YEAR TO MONTH</p> <p>Dengan indikasi interval_day_to_second: INTERVAL DAY TO SECOND</p> <p>Jika panjang > 4000: CLOB</p> <p>Selain itu gunakan: VARCHAR2 (length)</p>
UINT1	NUMBER (3)
UINT2	NUMBER (5)
UINT4	NUMBER (10)
UINT8	NUMBER (19)

AWS DMS tipe data	Tipe data Oracle
WSTRING	<p>Jika panjang > 2000: NCLOB</p> <p>Selain itu gunakan: NVARCHAR2 (length)</p>
BLOB	<p>BLOB</p> <p>Untuk menggunakan tipe data ini AWS DMS, Anda harus mengaktifkan penggunaan BLOB untuk tugas tertentu. Tipe data BLOB hanya didukung dalam tabel yang memiliki kunci primer</p>
CLOB	<p>CLOB</p> <p>Untuk menggunakan tipe data ini AWS DMS, Anda harus mengaktifkan penggunaan CLOB untuk tugas tertentu. Selama perubahan pengambilan data (CDC), tipe data CLOB didukung hanya dalam tabel yang memiliki kunci primer.</p> <p>STRING</p> <p>Tipe data Oracle VARCHAR2 pada sumber dengan ukuran yang dinyatakan lebih besar dari 4000 byte memetakan melalui AWS DMS CLOB ke STRING pada target Oracle.</p>
NCLOB	<p>NCLOB</p> <p>Untuk menggunakan tipe data ini AWS DMS, Anda harus mengaktifkan penggunaan NClobs untuk tugas tertentu. Selama CDC berjalan, tipe data NCLOB hanya didukung di dalam tabel yang mencakup kunci primer.</p> <p>WSTRING</p> <p>Tipe data Oracle VARCHAR2 pada sumber dengan ukuran yang dinyatakan lebih besar dari 4000 byte memetakan melalui AWS DMS NCLOB ke WSTRING pada target Oracle.</p>

AWS DMS tipe data	Tipe data Oracle
XMLTYPE	<p>tipe data target XMLTYPE hanya relevan dalam tugas replikasi Oracle-to-Oracle.</p> <p>Ketika Oracle digunakan sebagai sumber basis data, tipe data sumber direplikasi apa adanya pada target Oracle. Contohnya sebuah tipe data XMLTYPE pada sumber dibuat sebagai tipe data XMLTYPE pada target.</p>

Menggunakan basis data Microsoft SQL Server sebagai target untuk AWS Database Migration Service

Anda dapat memigrasi data ke basis data Microsoft SQL Server menggunakan AWS DMS. Dengan basis data SQL Server sebagai target, Anda dapat memigrasi data dari basis data SQL Server lain atau salah satu basis data didukung lainnya.

Untuk informasi tentang versi SQL Server yang AWS DMS mendukung sebagai target, lihat [Target untuk AWS DMS](#).

AWS DMS mendukung edisi lokal dan Amazon RDS Enterprise, Standard, Workgroup, dan Developer.

Untuk detail tambahan tentang bekerja dengan AWS DMS dan basis data target SQL Server, lihat berikut ini.

Topik

- [Keterbatasan menggunakan SQL Server sebagai target untuk AWS Database Migration Service](#)
- [Persyaratan keamanan saat menggunakan SQL Server sebagai target untuk AWS Database Migration Service](#)
- [Pengaturan titik akhir saat menggunakan SQL Server sebagai target AWS DMS](#)
- [Jenis data target untuk Microsoft SQL Server](#)

Keterbatasan menggunakan SQL Server sebagai target untuk AWS Database Migration Service

Batasan berikut berlaku saat menggunakan basis data SQL Server sebagai sumber untuk AWS DMS:

- Ketika Anda membuat tabel target SQL Server secara manual dengan kolom terkomputasi, maka penggunaan utilitas salinan massal BCP tidak mendukung replikasi beban penuh. Untuk menggunakan replikasi beban penuh, nonaktifkan pemuatan BCP dengan menyetel atribut koneksi tambahan (ECA) 'useBCPFullLoad=false' pada titik akhir. Untuk informasi tentang pengaturan ECA pada titik akhir, lihat [Membuat titik akhir sumber dan target](#) Untuk informasi lebih lanjut tentang bekerja menggunakan BCP, lihat [dokumentasi Microsoft SQL Server](#).
- Ketika mereplikasi tabel dengan tipe data spasial dari SQL Server (GEOMETRY dan GEOGRAPHY) AWS DMS mengganti setiap pengenal referensi spasial (SRID) yang mungkin telah Anda masukkan dengan SRID default. SRID default adalah 0 untuk GEOMETRY dan 4326 untuk GEOGRAPHY.
- Tidak mendukung tabel sementara. Migrasi tabel temporal dapat bekerja dengan tugas yang hanya terdiri dari replikasi saja dalam mode penerapan transaksional jika tabel tersebut dibuat secara manual pada target.
- Saat ini tipe data boolean dalam sumber PostgreSQL dimigrasi ke target SQLServer sebagai tipe data bit dengan nilai yang tidak konsisten.

Sebagai solusinya, lakukan hal berikut:

- Buat tabel dengan tipe VARCHAR(1) data untuk kolom (atau biarkan AWS DMS membuat tabel). Kemudian buatlah pemrosesan hilir memperlakukan "F" sebagai False dan "T" sebagai True.
- Untuk menghindari keharusan mengubah pemrosesan hilir, tambahkan aturan transformasi ke tugas untuk mengubah nilai "F" menjadi "0" dan "T" menjadi 1, dan simpan sebagai tipe data bit server SQL.
- AWS DMS tidak mendukung pemrosesan perubahan untuk mengatur nullabilitas kolom (menggunakan ALTER COLUMN [SET|DROP] NOT NULL klausa dengan pernyataan). ALTER TABLE
- Autentikasi Windows tidak didukung.

Persyaratan keamanan saat menggunakan SQL Server sebagai target untuk AWS Database Migration Service

Berikut penjelasan persyaratan keamanan untuk penggunaan AWS DMS dengan target Microsoft SQL Server:

- Parameter akun pengguna AWS DMS harus memiliki setidaknya peran pengguna `db_owner` di basis data SQL Server yang Anda hubungkan.
- Administrator sistem SQL Server harus memberikan izin ini untuk semua akun pengguna AWS DMS.

Pengaturan titik akhir saat menggunakan SQL Server sebagai target AWS DMS

Anda dapat menggunakan pengaturan endpoint untuk mengkonfigurasi database target SQL Server Anda mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir target menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--microsoft-sql-server-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan endpoint yang dapat Anda gunakan dengan SQL Server sebagai target.

Nama	Penjelasan
<code>ControlTablesFileGroup</code>	<p>Tentukan kelompok file untuk tabel internal AWS DMS. Ketika tugas replikasi dimulai, semua tabel kendali AWS DMS internal (<code>awsdms_apply_exception</code>, <code>awsdms_apply</code>, <code>awsdms_changes</code>) dibuat pada kelompok file yang sudah ditentukan.</p> <p>Nilai default: tidak ada</p> <p>Nilai valid: String</p> <p>Contoh: <code>--microsoft-sql-server-settings '{"ControlTablesFileGroup": "filegroup1"}'</code></p>

Nama	Penjelasan
	<p>Berikut ini adalah contoh perintah untuk membuat kelompok file.</p> <pre data-bbox="690 331 1507 926">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</pre>
ExecuteTimeout	<p>Gunakan atribut koneksi tambahan (ECA) ini untuk mengatur batas waktu pernyataan klien untuk instance SQL Server, dalam hitungan detik. Nilai default adalah 60 detik.</p> <p>Contoh: '{"ExecuteTimeout": 100}'</p>
UseBCPFullLoad	<p>Gunakan ini untuk mengatributkan transfer data untuk operasi beban penuh menggunakan BCP. Ketika tabel target berisi identitas kolom yang tidak ada di tabel sumber, Anda harus nonaktifkan pilihan Gunakan BCP untuk memuat tabel.</p> <p>Nilai default: benar</p> <p>Nilai yang benar: benar/salah</p> <p>Contoh: --microsoft-sql-server-settings '{"UseBCPFullLoad": false}'</p>

Jenis data target untuk Microsoft SQL Server

Tabel berikut menunjukkan tipe data target Microsoft SQL Server yang didukung saat menggunakan AWS DMS dan pemetaan default dari tipe data AWS DMS. Untuk informasi tambahan tentang tipe data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Tipe data AWS DMS	Tipe data SQL Server
BOOLEAN	TINYINT
BYTES	VARBINARY(length)
DATE	Untuk SQL Server 2008 dan yang lebih tinggi, gunakan DATE. Untuk versi sebelumnya, jika skalanya 3 atau kurang, gunakan DATETIME. Selain itu gunakan VARCHAR (37).
WAKTU	Untuk SQL Server 2008 dan yang lebih tinggi, gunakan DATETIME2 (%d). Untuk versi sebelumnya, jika skalanya 3 atau kurang, gunakan DATETIME. Selain itu gunakan VARCHAR (37).
DATETIME	Untuk SQL Server 2008 dan yang lebih tinggi, gunakan DATETIME2 (skala). Untuk versi sebelumnya, jika skalanya 3 atau kurang, gunakan DATETIME. Selain itu gunakan VARCHAR (37).
INT1	SMALLINT
INT2	SMALLINT
INT4	INT
INT8	BIGINT
NUMERIC	NUMERIC (p,s)
REAL4	REAL

Tipe data AWS DMS	Tipe data SQL Server
REAL8	FLOAT
STRING	<p>Jika kolom berupa kolom tanggal atau waktu, lakukan hal berikut:</p> <ul style="list-style-type: none"> • Untuk SQL Server 2008 dan yang lebih tinggi, gunakan DATETIME2. • Untuk versi sebelumnya, jika skalanya 3 atau kurang, gunakan DATETIME. Selain itu gunakan VARCHAR (37). <p>Jika kolom bukan berupa kolom tanggal atau waktu, gunakan VARCHAR (length).</p>
UINT1	TINYINT
UINT2	SMALLINT
UINT4	INT
UINT8	BIGINT
WSTRING	NVARCHAR (length)
BLOB	<p>VARBINARY(max)</p> <p>IMAGE</p> <p>Untuk menggunakan tipe data ini dengan AWS DMS, Anda harus mengaktifkan penggunaan BLOBs untuk tugas tertentu. AWS DMS mendukung tipe data BLOB hanya dalam tabel dengan kunci primer.</p>

Tipe data AWS DMS	Tipe data SQL Server
CLOB	<p>VARCHAR(max)</p> <p>Untuk menggunakan tipe data ini dengan AWS DMS, Anda harus mengaktifkan penggunaan CLOB untuk tugas tertentu. Selama perubahan penangkapan data (CDC) berjalan, AWS DMS mendukung tipe data CLOB hanya dalam tabel dengan kunci primer.</p>
NCLOB	<p>NVARCHAR(max)</p> <p>Untuk menggunakan tipe data ini dengan AWS DMS, Anda harus mengaktifkan penggunaan NCLOB untuk tugas tertentu. Selama CDC berjalan, AWS DMS mendukung tipe data NCLOB hanya dalam tabel yang mencakup kunci primer.</p>

Menggunakan basis data PostgreSQL sebagai target untuk AWS Database Migration Service

Anda dapat memigrasikan data ke database PostgreSQL AWS DMS menggunakan, baik dari database PostgreSQL lain atau dari salah satu database lain yang didukung.

Untuk informasi tentang versi PostgreSQL AWS DMS yang mendukung sebagai target, lihat. [Target untuk AWS DMS](#)

AWS DMS versi untuk digunakan

Gunakan AWS DMS versi apa pun yang tersedia.

Gunakan AWS DMS versi 3.4.3 dan lebih tinggi.

AWS DMS versi untuk digunakan

Gunakan AWS DMS versi 3.4.7 dan lebih tinggi.

Gunakan AWS DMS versi 3.5.1 dan lebih tinggi.

Note

- Amazon Aurora Serverless tersedia sebagai TARGET untuk Amazon Aurora dengan kompatibilitas PostgreSQL. Untuk informasi selengkapnya tentang Amazon Aurora Tanpa Server, lihat Menggunakan Amazon [Aurora Tanpa Server v2 di Panduan Pengguna Amazon Aurora](#).
- Kluster DB Aurora Nirserver hanya dapat diakses dari Amazon VPC dan tidak dapat menggunakan [alamat IP publik](#). Jadi jika Anda ingin memiliki instans replikasi di wilayah yang berbeda dari Aurora PostgreSQL Nirserver, Anda harus mengonfigurasi [peering VPC](#). Jika tidak, periksa ketersediaan [Wilayah](#) Aurora PostgreSQL Nirserver, dan putuskan untuk menggunakan salah satu wilayah tersebut untuk Aurora PostgreSQL Nirserver dan instans replikasi Anda.
- Kemampuan Babelfish dibangun ke Amazon Aurora dan tidak memiliki biaya tambahan. Untuk informasi lebih lanjut, lihat [Menggunakan Babelfish untuk Aurora PostgreSQL](#) sebagai target. AWS Database Migration Service

AWS DMS mengambil table-by-table pendekatan saat memigrasikan data dari sumber ke target dalam fase Full Load. Urutan tabel selama fase beban penuh tidak terjamin. Tabel tidak sinkron selama fase beban penuh dan saat penerapan transaksi yang di-cache untuk masing-masing tabel. Akibatnya, kendala integritas referensial aktif dapat mengakibatkan gagal tugas selama fase beban penuh.

Dalam PostgreSQL, foreign key (kendala integritas referensial) diimplementasikan menggunakan pemicu. Selama fase beban penuh, AWS DMS muat setiap tabel satu per satu. Kami sangat menyarankan Anda nonaktifkan pembatasan kunci asing selama beban penuh dengan menggunakan salah satu metode berikut:

- Nonaktifkan sementara semua pemicu dari instans, dan selesaikan beban penuh.
- Gunakan parameter `session_replication_role` di PostgreSQL.

Sewaktu-waktu, pemicu dapat berada di salah satu kondisi berikut: `origin`, `replica`, `always`, atau `disabled`. Saat parameter `session_replication_role` diatur ke `replica`, hanya pemicu dalam kondisi `replica` yang aktif, dan akan dipicu saat dipanggil. Jika tidak, pemicu tetap tidak aktif.

PostgreSQL memiliki mekanisme failsafe untuk mencegah tabel terpotong, bahkan ketika `session_replication_role` sudah diatur. Anda dapat menggunakan ini sebagai alternatif untuk menonaktifkan pemicu, untuk membantu menjalankan beban penuh hingga selesai. Untuk melakukannya, tetapkan mode persiapan tabel target untuk `DO_NOTHING`. Jika tidak, operasi `DROP` dan `TRUNCATE` gagal ketika ada pembatasan kunci asing.

Di Amazon RDS, Anda dapat mengontrol pengaturan parameter ini menggunakan grup parameter. Anda dapat mengatur parameter secara langsung untuk instans PostgreSQL yang berjalan di Amazon EC2.

Untuk detail tambahan tentang bekerja dengan database PostgreSQL sebagai target AWS DMS, lihat bagian berikut:

Topik

- [Batasan menggunakan PostgreSQL sebagai target untuk AWS Database Migration Service](#)
- [Persyaratan keamanan saat menggunakan database PostgreSQL sebagai target AWS Database Migration Service](#)
- [Pengaturan titik akhir dan Atribut Koneksi Ekstra \(ECA\) saat menggunakan PostgreSQL sebagai target AWS DMS](#)
- [Tipe data target untuk PostgreSQL](#)
- [Menggunakan Babelfish untuk Aurora PostgreSQL sebagai target untuk AWS Database Migration Service](#)

Batasan menggunakan PostgreSQL sebagai target untuk AWS Database Migration Service

Batasan berikut berlaku ketika menggunakan basis data PostgreSQL sebagai target untuk AWS DMS:

- Untuk migrasi heterogen, tipe data JSON dikonversi ke tipe data CLOB Asli secara internal.
- Dalam migrasi Oracle ke PostgreSQL, jika kolom di Oracle berisi karakter NULL (nilai hex U+0000), mengubah karakter NULL menjadi spasi (nilai hex U+0020) AWS DMS . Hal ini disebabkan oleh keterbatasan PostgreSQL.
- AWS DMS tidak mendukung replikasi ke tabel dengan indeks unik yang dibuat dengan fungsi coalesce.
- Jika tabel Anda menggunakan urutan, perbarui nilai NEXTVAL untuk setiap urutan dalam database target setelah Anda menghentikan replikasi dari database sumber. AWS DMS menyalin data dari database sumber Anda, tetapi tidak memigrasikan urutan ke target selama replikasi yang sedang berlangsung.

Persyaratan keamanan saat menggunakan database PostgreSQL sebagai target AWS Database Migration Service

Untuk tujuan keamanan, akun pengguna yang digunakan untuk migrasi data harus merupakan pengguna terdaftar di basis data PostgreSQL yang Anda gunakan sebagai target.

Titik akhir target PostgreSQL Anda memerlukan izin pengguna minimum untuk menjalankan AWS DMS migrasi, lihat contoh berikut.

```
CREATE USER newuser WITH PASSWORD 'your-password';  
ALTER SCHEMA schema_name OWNER TO newuser;
```

Atau,

```
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;
```

Pengaturan titik akhir dan Atribut Koneksi Ekstra (ECA) saat menggunakan PostgreSQL sebagai target AWS DMS

Anda dapat menggunakan pengaturan endpoint dan Extra Connection Attributes (ECA) untuk mengonfigurasi database target PostgreSQL Anda.

Anda menentukan pengaturan saat Anda membuat titik akhir target menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--postgresql-settings '{"EndpointSetting": "value", ...}'` JSON.

Anda menentukan ECA menggunakan `ExtraConnectionAttributes` parameter untuk titik akhir Anda.

Tabel berikut menunjukkan pengaturan endpoint yang dapat Anda gunakan dengan PostgreSQL sebagai target.

Nama	Deskripsi
<code>MaxFileSize</code>	<p>Menentukan ukuran maksimum (dalam KB) dari setiap file .csv yang digunakan untuk mentransfer data ke PostgreSQL.</p> <p>Nilai default: 32.768 KB (32 MB)</p> <p>Nilai valid: 1-1.048.576 KB (hingga 1,1 GB)</p> <p>Contoh: <code>--postgresql-settings '{"MaxFileSize": 512}'</code></p>
<code>ExecuteTimeout</code>	<p>Menetapkan pernyataan timeout klien untuk instans PostgreSQL, dalam hitungan detik. Nilai default adalah 60 detik.</p> <p>Contoh: <code>--postgresql-settings '{"ExecuteTimeout": 100}'</code></p>
<code>AfterConnectScript= SET session_replication_role = replica</code>	<p>Atribut ini memiliki AWS DMS bypass kunci asing dan pemicu pengguna untuk mengurangi waktu yang dibutuhkan untuk memuat data secara massal.</p>

Nama	Deskripsi
MapUnboundedNumericAsString	<p>Parameter ini memperlakukan kolom dengan tipe data NUMERIC tak terbatas sebagai STRING agar berhasil bermigrasi tanpa kehilangan presisi nilai numerik. Gunakan parameter ini hanya untuk replikasi dari sumber PostgreSQL ke target PostgreSQL, atau basis data dengan kompatibilitas PostgreSQL.</p> <p>Nilai default: SALAH</p> <p>Nilai valid: SALAH/BETUL</p> <p>Contoh: <code>--postgre-sql-settings '{"MapUnboundedNumericAsString": "true"}</code></p> <p>Menggunakan parameter ini dapat mengakibatkan beberapa degradasi performa replikasi karena transformasi dari numerik ke string dan kembali ke numerik. Parameter ini didukung untuk digunakan oleh DMS versi 3.4.4 dan versi yang lebih tinggi</p> <div data-bbox="688 1100 1507 1753" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p> Note</p><p>Hanya gunakan MapUnboundedNumericAsString di sumber PostgreSQL dan titik akhir target bersama-sama. Penggunaan titik akhir PostgreSQL sumber membatasi presisi hingga 28 selama CDC. MapUnboundedNumericAsString Penggunaan MapUnboundedNumericAsString pada titik akhir target, memigrasikan data dengan Precision 28 Scale 6. Jangan gunakan MapUnboundedNumericAsString dengan target non-PostgreSQL.</p></div>

Nama	Deskripsi
loadUsingCSV	<p>Gunakan Extra Connection Attribute (ECA) ini untuk mentransfer data untuk operasi full-load menggunakan perintah \ COPY.</p> <p>Nilai default: true</p> <p>Nilai yang benar: benar/salah</p> <p>Contoh ECA: loadUsingCSV=true;</p> <p>Catatan: Menyetel ECA ini ke false dapat mengakibatkan beberapa degradasi kinerja replikasi karena INSERT dijalankan secara langsung.</p>
DatabaseMode	<p>Gunakan atribut ini untuk mengubah perilaku default penanganan replikasi titik akhir yang kompatibel dengan Postgresql yang memerlukan beberapa konfigurasi tambahan, seperti titik akhir Babelfish.</p> <p>Nilai default: DEFAULT</p> <p>Nilai valid: DEFAULT, BABELFISH</p> <p>Contoh: DatabaseMode=default;</p>
BabelfishDatabaseName	<p>Gunakan atribut ini untuk menentukan nama database target Babelfish T-SQL yang sedang dimigrasi. Ini wajib diisi jika DatabaseMode diatur ke Babelfish . Ini bukan babelfish_db database yang dicadangkan.</p> <p>Contoh: BabelfishDatabaseName=TargetDb;</p>

Tipe data target untuk PostgreSQL

Titik akhir database PostgreSQL untuk AWS DMS mendukung sebagian besar tipe data database PostgreSQL. Tabel berikut menunjukkan tipe data target database PostgreSQL yang didukung saat AWS DMS menggunakan dan pemetaan default dari tipe data. AWS DMS

Untuk informasi tambahan tentang tipe AWS DMS data, lihat [Tipe data untuk AWS Database Migration Service](#).

AWS DMS tipe data	Tipe data PostgreSQL
BOOLEAN	BOOLEAN
BLOB	BYTEA
BYTES	BYTEA
DATE	DATE
TIME	TIME
DATETIME	Jika skala dari 0 sampai 6, gunakan TIMESTAMP. Jika skala dari 7 sampai 9, gunakan VARCHAR (37).
INT1	SMALLINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	DECIMAL (P,S)
REAL4	FLOAT4
REAL8	FLOAT8
STRING	Jika panjangnya dari 1 sampai 21.845, gunakan VARCHAR (panjang dalam bytes). Jika panjangnya 21.846 hingga 2.147.483.647, gunakan VARCHAR (65535).
UINT1	SMALLINT
UINT2	INTEGER

AWS DMS tipe data	Tipe data PostgreSQL
UINT4	BIGINT
UINT8	BIGINT
WSTRING	Jika panjangnya dari 1 sampai 21.845, gunakan VARCHAR (panjang dalam bytes). Jika panjangnya 21.846 hingga 2.147.483.647, gunakan VARCHAR (65535).
NCLOB	TEXT
CLOB	TEXT

Note

Saat mereplikasi dari sumber PostgreSQL AWS DMS, buat tabel target dengan tipe data yang sama untuk semua kolom, selain kolom dengan tipe data yang ditentukan pengguna. Dalam kasus seperti itu, tipe data dibuat sebagai “karakter bervariasi” dalam target.

Menggunakan Babelfish untuk Aurora PostgreSQL sebagai target untuk AWS Database Migration Service

Anda dapat memigrasikan tabel sumber SQL Server ke target Babelfish untuk Amazon Aurora PostgreSQL menggunakan AWS Database Migration Service Dengan Babelfish, Aurora PostgreSQL memahami T-SQL, dialek SQL milik Microsoft SQL Server, dan mendukung protokol komunikasi yang sama. Jadi, aplikasi yang ditulis untuk SQL Server sekarang dapat bekerja dengan Aurora dengan perubahan kode yang lebih sedikit. Kemampuan Babelfish dibangun ke Amazon Aurora dan tidak memiliki biaya tambahan. Anda dapat mengaktifkan Babelfish di cluster Amazon Aurora Anda dari konsol Amazon RDS.

Saat Anda membuat titik akhir AWS DMS target menggunakan perintah AWS DMS konsol, API, atau CLI, tentukan mesin target sebagai Amazon Aurora PostgreSQL, dan beri nama database, `babelfish_db`. Di bagian Pengaturan Endpoint, tambahkan pengaturan untuk mengatur ke

Babelfish dan DatabaseMode BabelfishDatabaseName ke nama database target Babelfish T-SQL.

Menambahkan aturan transformasi ke tugas migrasi Anda

Saat Anda menentukan tugas migrasi untuk target Babelfish, Anda perlu menyertakan aturan transformasi yang memastikan DMS menggunakan tabel T-SQL Babelfish yang telah dibuat sebelumnya dalam database target.

Pertama, tambahkan aturan transformasi ke tugas migrasi Anda yang membuat semua nama tabel menjadi huruf kecil. Babelfish menyimpan sebagai huruf kecil dalam `pg_class` katalog PostgreSQL nama-nama tabel yang Anda buat menggunakan T-SQL. Namun, ketika Anda memiliki tabel SQL Server dengan nama kasus campuran, DMS membuat tabel menggunakan tipe data asli PostgreSQL, bukan tipe data yang kompatibel dengan T-SQL. Untuk alasan itu, pastikan untuk menambahkan aturan transformasi yang membuat semua nama tabel huruf kecil. Perhatikan bahwa nama kolom tidak boleh diubah menjadi huruf kecil.

Selanjutnya, jika Anda menggunakan mode migrasi multidatabase saat menentukan klaster, tambahkan aturan transformasi yang mengganti nama skema SQL Server asli. Pastikan untuk mengganti nama nama skema SQL Server untuk menyertakan nama database T-SQL. Misalnya, jika nama skema SQL Server asli adalah `dbo`, dan nama database T-SQL `Andamydb`, ganti nama skema menjadi menggunakan aturan transformasi `mydb_dbo`

Jika Anda menggunakan mode database tunggal, Anda tidak memerlukan aturan transformasi untuk mengganti nama nama skema. Nama skema memiliki one-to-one pemetaan dengan target database T-SQL di Babelfish.

Aturan transformasi sampel berikut membuat semua nama tabel huruf kecil, dan mengganti nama skema SQL Server asli dari `dbo` ke `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": []
  }
]
}

```

Keterbatasan untuk menggunakan titik akhir target PostgreSQL dengan tabel Babelfish

Batasan berikut berlaku saat menggunakan titik akhir target PostgreSQL dengan tabel Babelfish:

- Untuk mode persiapan tabel Target, gunakan hanya mode Do nothing atau Truncate. Jangan gunakan tabel Drop pada mode target. Dalam mode itu, DMS membuat tabel sebagai tabel PostgreSQL yang T-SQL mungkin tidak mengenali.
- AWS DMS tidak mendukung tipe data `sql_variant`.
- Babelfish tidak mendukung `HEIRARCHYID`, `GEOMETRY`, dan tipe `GEOGRAPHY` data. Untuk memigrasikan tipe data ini, Anda dapat menambahkan aturan transformasi untuk mengonversi tipe data. `wstring(250)`

- Babelfish hanya mendukung migrasi BINARY, VARBINARY, dan tipe IMAGE data menggunakan tipe data. BYTEA [Untuk versi Aurora PostgreSQL sebelumnya, Anda dapat menggunakan DMS untuk memigrasikan tabel ini ke titik akhir target Babelfish.](#) Anda tidak perlu menentukan panjang untuk tipe BYTEA data, seperti yang ditunjukkan pada contoh berikut.

```
[Picture] [VARBINARY](max) NULL
```

Ubah tipe data T-SQL sebelumnya ke tipe data yang didukung T-SQL. BYTEA

```
[Picture] BYTEA NULL
```

- Untuk versi Aurora PostgreSQL Babelfish sebelumnya, jika Anda membuat tugas migrasi untuk replikasi berkelanjutan dari SQL Server ke Babelfish menggunakan titik akhir target PostgreSQL, Anda perlu menetapkan tipe data ke tabel apa pun yang menggunakan kolom. SERIAL IDENTITY Dimulai dengan Aurora PostgreSQL (versi 15.3/14.8 dan lebih tinggi) dan Babelfish (versi 3.2.0 dan lebih tinggi), kolom identitas didukung, dan tidak lagi diperlukan untuk menetapkan tipe data SERIAL. Untuk informasi selengkapnya, lihat [Penggunaan SERIAL](#) di bagian Urutan dan Identitas dari SQL Server ke Aurora PostgreSQL Migration Playbook. Kemudian, saat Anda membuat tabel di Babelfish, ubah definisi kolom dari berikut ini.

```
[IDCo1] [INT] IDENTITY(1,1) NOT NULL PRIMARY KEY
```

Ubah yang sebelumnya menjadi berikut.

```
[IDCo1] SERIAL PRIMARY KEY
```

Aurora PostgreSQL yang kompatibel dengan Babelfish membuat urutan menggunakan konfigurasi default dan menambahkan batasan ke kolom. NOT NULL Urutan yang baru dibuat berperilaku seperti urutan biasa (bertambah 1) dan tidak memiliki opsi komposit SERIAL.

- Setelah memigrasikan data dengan tabel yang menggunakan IDENTITY kolom atau tipe SERIAL data, setel ulang objek urutan berbasis PostgreSQL berdasarkan nilai maksimum kolom. Setelah melakukan beban penuh tabel, gunakan kueri T-SQL berikut untuk menghasilkan pernyataan untuk menyemai objek urutan terkait.

```
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(%)';

```

Kueri menghasilkan serangkaian pernyataan SELECT yang Anda jalankan untuk memperbarui nilai IDENTITAS dan SERIAL maksimum.

- Untuk versi Babelfish sebelum 3.2, mode LOB Penuh dapat mengakibatkan kesalahan tabel. Jika itu terjadi, buat tugas terpisah untuk tabel yang gagal dimuat. Kemudian gunakan mode LOB Terbatas untuk menentukan nilai yang sesuai untuk ukuran LOB Maksimum (KB). Pilihan lain adalah untuk mengatur pengaturan SQL Server Endpoint Connection Attribute. `ForceFullLob=True`
- Untuk versi Babelfish sebelum 3.2, melakukan validasi data dengan tabel Babelfish yang tidak menggunakan kunci primer berbasis integer menghasilkan pesan bahwa kunci unik yang sesuai tidak dapat ditemukan. Dimulai dengan Aurora PostgreSQL (versi 15.3/14.8 dan lebih tinggi) dan Babelfish (versi 3.2.0 dan lebih tinggi), validasi data untuk kunci primer non-integer didukung.
- Karena perbedaan presisi dalam jumlah tempat desimal selama beberapa detik, DMS melaporkan kegagalan validasi data untuk tabel Babelfish yang menggunakan tipe data. DATETIME Untuk menekan kegagalan tersebut, Anda dapat menambahkan tipe aturan validasi berikut untuk tipe DATETIME data.

```

{
  "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"
}
```

Menggunakan basis data yang kompatibel dengan MySQL sebagai target untuk AWS Database Migration Service

Anda dapat memigrasikan data ke database yang kompatibel dengan MySQL menggunakan AWS DMS, dari salah satu mesin data sumber yang mendukung. AWS DMS Jika Anda bermigrasi ke database yang kompatibel dengan MySQL lokal, maka mesin sumber Anda AWS DMS harus berada di dalam ekosistem. AWS Mesin dapat menggunakan layanan yang AWS dikelola seperti Amazon RDS, Amazon Aurora, atau Amazon S3. Atau mesin bisa berada di basis data terkelola mandiri di Amazon EC2.

Anda dapat menggunakan SSL untuk mengenkripsi koneksi antara titik akhir yang kompatibel dengan MySQL dan instans replikasi. Untuk informasi lebih lanjut tentang penggunaan SSL dengan titik akhir yang kompatibel dengan MySQL, lihat [Menggunakan SSL dengan AWS Database Migration Service](#).

Untuk informasi tentang versi MySQL AWS DMS yang mendukung sebagai target, lihat [Target untuk AWS DMS](#)

Anda dapat menggunakan database yang kompatibel dengan MySQL berikut sebagai target untuk: 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

Terlepas dari mesin penyimpanan sumber yang digunakan (myISAM, MEMORY, dan sebagainya), AWS DMS menciptakan tabel target yang kompatibel dengan MySQL sebagai tabel InnoDB secara default.

Jika Anda membutuhkan tabel di mesin penyimpanan selain InnoDB, Anda dapat membuatnya secara manual pada target yang kompatibel dengan MySQL dan memigrasi tabel dengan memilih Tidak melakukan apapun. Untuk informasi lebih lanjut, lihat [Pengaturan tugas beban penuh](#).

Untuk detail tambahan tentang bekerja dengan basis data yang kompatibel dengan MySQL sebagai target AWS DMS, lihat bagian berikut.

Topik

- [Menggunakan database yang kompatibel dengan MySQL sebagai target untuk AWS Database Migration Service](#)
- [Batasan dalam menggunakan database yang kompatibel dengan MySQL sebagai target untuk AWS Database Migration Service](#)
- [Pengaturan titik akhir saat menggunakan database yang kompatibel dengan MySQL sebagai target AWS DMS](#)
- [Tipe data target untuk MySQL](#)

Menggunakan database yang kompatibel dengan MySQL sebagai target untuk AWS Database Migration Service

Sebelum Anda mulai bekerja dengan basis data yang kompatibel dengan MySQL sebagai target untuk AWS DMS, pastikan bahwa Anda telah memenuhi prasyarat berikut:

- Berikan akun pengguna AWS DMS yang memiliki hak membaca/menulis ke database yang kompatibel dengan MySQL. Untuk membuat hak yang diperlukan, jalankan perintah berikut.

```
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>'@'%' ;
```

- Selama fase migrasi beban penuh, Anda harus nonaktifkan kunci asing pada tabel target Anda. Untuk menonaktifkan pemeriksaan kunci asing pada database yang kompatibel dengan MySQL selama pemuatan penuh, Anda dapat menambahkan perintah berikut ke bagian Atribut koneksi tambahan AWS DMS konsol untuk titik akhir target Anda.

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

- Tetapkan parameter basis data `local_infile = 1` untuk mengaktifkan AWS DMS agar memuat data ke basis data target.

Batasan dalam menggunakan database yang kompatibel dengan MySQL sebagai target untuk AWS Database Migration Service

Saat menggunakan database MySQL sebagai target AWS DMS , tidak mendukung hal berikut:

- Beberapa pernyataan bahasa definisi data (DDL): TRUNCATE PARTITION, DROP TABLE, dan RENAME TABLE.
- Menggunakan pernyataan ALTER TABLE *table_name* ADD COLUMN *column_name* untuk menambahkan kolom ke bagian depan atau tengah tabel.
- Saat memuat data ke target yang kompatibel dengan MySQL dalam tugas pemuatan penuh, AWS DMS tidak melaporkan kesalahan yang disebabkan oleh kendala dalam log tugas, yang dapat menyebabkan kesalahan kunci duplikat atau ketidakcocokan dengan jumlah catatan. Hal ini disebabkan oleh cara MySQL menangani data lokal dengan perintah. LOAD DATA Pastikan untuk melakukan hal berikut selama fase beban penuh:
 - Nonaktifkan kendala
 - Gunakan AWS DMS validasi untuk memastikan data konsisten.
- Ketika Anda memperbarui nilai kolom ke nilai yang ada, basis data yang kompatibel dengan MySQL akan menampilkan Peringatan 0 rows affected. Meskipun secara teknis perilaku ini bukan merupakan kesalahan, penanganan situasi ini berbeda dibandingkan mesin basis data

lain. Misalnya Oracle melakukan update dari satu baris. Untuk database yang kompatibel dengan MySQL, buat entri di AWS DMS tabel kontrol `awsdms_apply_exceptions` dan mencatat peringatan berikut.

Some changes from the source database had no impact when applied to the target database. See `awsdms_apply_exceptions` table for details.

- Aurora Nirserver tersedia sebagai target untuk Amazon Aurora versi 2, kompatibel dengan MySQL versi 5.7. (Pilih Aurora MySQL versi 2.07.1 untuk dapat menggunakan Aurora Nirserver dengan kompatibilitas terhadap MySQL 5.7.) Untuk informasi selengkapnya tentang Aurora Tanpa Server, lihat Menggunakan [Aurora Tanpa Server v2 di Panduan Pengguna Amazon Aurora](#).
- AWS DMS tidak mendukung penggunaan titik akhir pembaca untuk Aurora atau Amazon RDS, kecuali instans dalam mode yang dapat ditulis, yaitu parameter `innodb_read_only` dan disetel `read_only` ke `0` atau `OFF`. Untuk informasi selengkapnya tentang penggunaan Amazon RDS dan Aurora sebagai target, lihat berikut ini:
 - [Menentukan instans DB mana yang terhubung dengan Anda](#)
 - [Memperbarui replika baca dengan MySQL](#)

Pengaturan titik akhir saat menggunakan database yang kompatibel dengan MySQL sebagai target AWS DMS

Anda dapat menggunakan pengaturan titik akhir untuk mengonfigurasi basis data target yang kompatibel dengan MySQL mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir target menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--my-sql-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan endpoint yang dapat Anda gunakan dengan MySQL sebagai target.

Nama	Deskripsi
TargetDbType	Tentukan tujuan untuk migrasi tabel sumber pada target, baik untuk basis data tunggal atau beberapa basis data.

Nama	Deskripsi
	<p>Jika Anda menentukan <code>SPECIFIC_DATABASE</code> , Anda perlu menentukan nama database, baik saat menggunakan AWS CLI atau AWS Management Console.</p> <p>Nilai default: <code>MULTIPLE_DATABASES</code></p> <p>Nilai valid: <code>{SPECIFIC_DATABASE , MULTIPLE_DATABASES }</code></p> <p>Contoh: <code>--my-sql-settings '{"TargetDbType": "MULTIPLE_DATABASES"}'</code></p>
ParallelLoadThreads	<p>Meningkatkan performa saat memuat data ke basis data target yang kompatibel dengan MySQL. Menentukan berapa banyak thread yang digunakan untuk memuat data ke dalam basis data target yang kompatibel dengan MySQL. Menetapkan sejumlah besar thread dapat memiliki efek yang berlawanan pada performa basis data, karena koneksi terpisah diperlukan untuk setiap utas.</p> <p>Nilai default: 1</p> <p>Nilai valid: 1-5</p> <p>Contoh: <code>--my-sql-settings '{"ParallelLoadThreads": 1}'</code></p>

Nama	Deskripsi
AfterConnectScript	<p>Menentukan script untuk menjalankan segera setelah AWS DMS terhubung ke titik akhir.</p> <p>Misalnya, Anda dapat menentukan bahwa target yang kompatibel dengan MySQL harus menerjemahkan pernyataan yang diterima ke dalam kumpulan karakter latin1, yang merupakan kumpulan karakter default yang dikompilasi dari database. Parameter ini biasanya meningkatkan performa ketika mengonversi dari client UTF8.</p> <p>Contoh: <code>--my-sql-settings '{"AfterConnectScript": "SET character_set_connection='latin1'"}</code></p>
MaxFileSize	<p>Menentukan ukuran maksimum (dalam KB) dari setiap file .csv yang digunakan untuk mentransfer data ke basis data yang kompatibel dengan MySQL.</p> <p>Nilai default: 32.768 KB (32 MB)</p> <p>Nilai valid: 1-1.048.576</p> <p><code>--my-sql-settings '{"MaxFileSize": 512}'</code></p>
CleanSrcMetadataOnMismatch	<p>Membersihkan dan membuat ulang informasi metadata tabel pada instans replikasi ketika terjadi ketidakcocokan. Contohnya adalah situasi ketika menjalankan pernyataan DDL 'alter' di tabel dapat menghasilkan informasi berbeda tentang tabel yang disimpan dalam cache di instans replikasi. Boolean.</p> <p>Nilai default: false</p> <p>Contoh: <code>--my-sql-settings '{"CleanSrcMetadataOnMismatch": false}'</code></p>

Anda juga dapat menggunakan atribut koneksi tambahan untuk mengonfigurasi basis data target yang kompatibel dengan MySQL Anda.

Tabel berikut menunjukkan atribut koneksi tambahan yang dapat Anda gunakan dengan MySQL sebagai target.

Nama	Penjelasan
<code>Initstmt=SET FOREIGN_KEY_CHECKS=0;</code>	<p>Menonaktifkan memeriksa kunci asing.</p> <p>Contoh: <code>--extra-connection-attributes "Initstmt=SET FOREIGN_KEY_CHECKS=0;"</code></p>
<code>Initstmt=SET time_zone</code>	<p>Menentukan zona waktu untuk target basis data yang kompatibel dengan MySQL.</p> <p>Nilai default: UTC</p> <p>Nilai yang valid: Nama zona waktu yang tersedia di database MySQL target.</p> <p>Contoh: <code>--extra-connection-attributes "Initstmt=SET time_zone= <i>US/Pacific</i> ;"</code></p>

Atau, Anda dapat menggunakan `AfterConnectScript` parameter `--my-sql-settings` perintah untuk menonaktifkan pemeriksaan kunci asing dan menentukan zona waktu untuk database Anda.

Tipe data target untuk MySQL

Tabel berikut menunjukkan tipe data target database MySQL yang didukung saat AWS DMS menggunakan dan pemetaan AWS DMS default dari tipe data.

Untuk informasi tambahan tentang tipe AWS DMS data, lihat [Tipe data untuk AWS Database Migration Service](#).

AWS DMS tipe data	Tipe data MySQL
BOOLEAN	BOOLEAN

AWS DMS tipe data	Tipe data MySQL
BYTES	Jika panjangnya dari 1 sampai 65.535, maka gunakan VARBINARY (length). Jika panjangnya dari 65.536 hingga 2.147.483.647 maka gunakan LONGLOB.
DATE	DATE
TIME	TIME
TIMESTAMP	"If scale => 0 and =< 6, then: DATETIME (Scale) If scale is => 7 and =< 9, then: VARCHAR (37)"
INT1	TINYINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	DECIMAL (p,s)
REAL4	FLOAT
REAL8	DOUBLE PRECISION
STRING	Jika panjangnya dari 1 sampai 21.845 maka gunakan VARCHAR (panjang). Jika panjangnya dari 21.846 hingga 2.147.483.647 maka gunakan LONGTEXT.
UINT1	UNSIGNED TINYINT
UINT2	UNSIGNED SMALLINT

AWS DMS tipe data	Tipe data MySQL
UINT4	UNSIGNED INTEGER
UINT8	UNSIGNED BIGINT
WSTRING	<p>Jika panjangnya dari 1 sampai 32.767 maka gunakan VARCHAR (panjang).</p> <p>Jika panjangnya dari 32.768 hingga 2.147.483.647 maka gunakan LONGTEXT.</p>
BLOB	<p>Jika panjangnya dari 1 sampai 65.535 maka gunakan BLOB.</p> <p>Jika panjangnya dari 65.536 hingga 2.147.483.647 maka gunakan LONGBLOB.</p> <p>Jika panjangnya 0, gunakan LONGBLOB (support LOB penuh).</p>
NCLOB	<p>Jika panjangnya dari 1 sampai 65.535, gunakan TEXT.</p> <p>Jika panjangnya dari 65.536 hingga 2.147.483.647, gunakan LONGTEXT dengan ucs2 untuk CHARACTER SET.</p> <p>Jika panjangnya 0, gunakan LONGTEXT (support LOB penuh) dengan ucs2 untuk CHARACTER SET.</p>
CLOB	<p>Jika panjangnya dari 1 sampai 65.535, gunakan TEXT.</p> <p>Jika panjangnya dari 65.536 hingga 2.147.483.647, gunakan LONGTEXT.</p> <p>Jika panjangnya 0, gunakan LONGTEXT (supportLOB penuh).</p>

Menggunakan basis data Amazon Redshift sebagai target untuk AWS Database Migration Service

Anda dapat melakukan migrasi data ke basis data Amazon Redshift menggunakan AWS Database Migration Service. Amazon Redshift adalah layanan gudang data dengan skala petabyte yang dikelola penuh di cloud. Dengan basis data Amazon Redshift sebagai target, Anda dapat memigrasi data dari semua basis data sumber lain yang didukung.

Anda dapat menggunakan Amazon Redshift Serverless sebagai target. AWS DMS Untuk informasi lebih lanjut, lihat [Menggunakan AWS DMS Amazon Redshift Serverless sebagai Target](#) berikut ini.

Klaster Amazon Redshift harus berada di akun AWS yang sama dan wilayah AWS yang sama sebagai instans replikasi.

Selama migrasi basis data ke Amazon Redshift, pertama-tama AWS DMS memindahkan data ke Amazon S3 bucket. Ketika file berada di bucket Amazon S3, AWS DMS mentransfer mereka ke tabel yang tepat di gudang data Amazon Redshift. AWS DMS menciptakan bucket S3 di wilayah AWS yang sama dengan basis data Amazon Redshift. Instans replikasi AWS DMS harus ditempatkan di wilayah AWS yang sama.

Jika Anda menggunakan AWS CLI atau DMS API untuk memigrasikan data ke Amazon Redshift, siapkan AWS Identity and Access Management (IAM) role untuk mengizinkan akses S3. Untuk informasi lebih lanjut tentang pembuatan IAM role, lihat [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#).

Titik akhir Amazon Redshift menyediakan otomatisasi penuh untuk hal berikut:

- Pembuatan skema dan pemetaan tipe data
- Beban penuh dari tabel basis data sumber
- Beban tambahan dari perubahan yang dibuat ke tabel sumber
- Penerapan perubahan skema dalam bahasa definisi data (DDL) yang dibuat ke tabel sumber
- Sinkronisasi antara proses beban penuh dan penangkapan perubahan data (CDC).

AWS Database Migration Service mendukung operasi beban penuh dan perubahan pemrosesan. AWS DMS membaca data dari basis data sumber dan menciptakan serangkaian file comma-separated value(.csv). Untuk operasi beban penuh, AWS DMS menciptakan file untuk setiap tabel. AWS DMS kemudian menyalin file tabel untuk setiap tabel ke folder terpisah di Amazon S3. Ketika file diunggah ke Amazon S3, AWS DMS mengirimkan perintah salin, kemudian data dalam file disalin

ke Amazon Redshift. Untuk operasi perubahan-pemrosesan, AWS DMS menyalin net changes ke file .csv. Kemudian AWS DMS mengunggah file net changes ke Amazon S3 dan menyalin data ke Amazon Redshift.

Untuk detail tambahan tentang bekerja dengan Amazon Redshift sebagai target untuk AWS DMS, lihat bagian berikut:

Topik

- [Prasyarat untuk menggunakan basis data Amazon Redshift sebagai target untuk AWS Database Migration Service](#)
- [Hak yang diperlukan untuk menggunakan Redshift sebagai target](#)
- [Keterbatasan penggunaan Amazon Redshift sebagai target untuk AWS Database Migration Service](#)
- [Mengonfigurasi basis data Amazon Redshift sebagai target untuk AWS Database Migration Service](#)
- [Menggunakan perutean VPC yang disempurnakan dengan Amazon Redshift sebagai target AWS Database Migration Service](#)
- [Membuat dan menggunakan kunci AWS KMS untuk mengenkripsi data target Amazon Redshift](#)
- [Pengaturan titik akhir saat menggunakan Amazon Redshift sebagai target untuk AWS DMS](#)
- [Menggunakan kunci enkripsi data, dan bucket Amazon S3 sebagai penyimpanan perantara](#)
- [Pengaturan tugas multithreaded untuk Amazon Redshift](#)
- [Tipe data target untuk Amazon Redshift](#)
- [Menggunakan AWS DMS Amazon Redshift Serverless sebagai Target](#)

Prasyarat untuk menggunakan basis data Amazon Redshift sebagai target untuk AWS Database Migration Service

Daftar berikut menjelaskan prasyarat yang diperlukan untuk bekerja dengan Amazon Redshift sebagai target migrasi data:

- Gunakan konsol manajemen AWS untuk meluncurkan klaster Amazon Redshift. Catat informasi dasar tentang akun AWS dan klaster Amazon Redshift Anda, seperti kata sandi, nama pengguna, dan nama basis data. Anda perlu nilai-nilai ini saat membuat titik akhir target Amazon Redshift target.
- Klaster Amazon Redshift harus berada di akun AWS dan Wilayah AWS yang sama dengan instans replikasi.

- Instans replikasi AWS DMS membutuhkan konektivitas jaringan ke titik akhir Amazon Redshift (nama host dan port) yang digunakan oleh klaster Anda.
- AWS DMS Menggunakan bucket Amazon S3 untuk mentransfer data ke basis data Amazon Redshift. Agar AWS DMS dapat membuat bucket, konsol tersebut menggunakan IAM role, `dms-access-for-endpoint`. Jika Anda menggunakan AWS CLI atau DMS API untuk membuat migrasi basis data dengan Amazon Redshift sebagai basis data target, Anda harus membuat IAM role ini. Untuk informasi lebih lanjut tentang pembuatan peran, lihat [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#).
- AWS DMS mengonversi BLOB, CLOB, dan NCLOB ke VARCHAR pada instans target Amazon Redshift. Amazon Redshift tidak support tipe data VARCHAR yang lebih besar dari 64 KB, sehingga Anda tidak dapat menyimpan LOB tradisional di Amazon Redshift.
- Tetapkan pengaturan tugas metadata target `BatchApplyEnabled` ke `true` for AWS DMS untuk menangani perubahan pada tabel target Amazon Redshift selama CDC. Diperlukan sebuah Kunci Primer pada sumber dan target tabel. Tanpa Kunci Primer, perubahan diterapkan pernyataan demi pernyataan. Dan hal itu dapat memengaruhi performa tugas selama CDC berlangsung dengan menyebabkan latensi target dan memengaruhi antrean melakukan klaster.

Hak yang diperlukan untuk menggunakan Redshift sebagai target

Gunakan perintah GRANT untuk menentukan hak akses pengguna atau grup pengguna. Hak termasuk pilihan akses seperti dapat membaca data dalam tabel dan view, menulis data, dan membuat tabel. Untuk informasi lebih lanjut tentang penggunaan GRANT dengan Amazon Redshift, lihat [GRANT](#) di Panduan Developer Basis data Amazon Redshift.

Berikut adalah sintaks untuk memberikan hak khusus untuk tabel, basis data, skema, fungsi, prosedur, atau hak tingkat-bahasa pada tabel dan view di 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 } [, ...]

```

Berikut adalah sintaks untuk hak tingkat-kolom pada tabel dan views di Amazon Redshift.

```

GRANT { { SELECT | UPDATE } ( column_name [, ...] ) [, ...] | ALL [ PRIVILEGES ]
( column_name [, ...] ) }
    ON { [ TABLE ] table_name [, ...] }
    TO { username | GROUP group_name | PUBLIC } [, ...]

```

Berikut adalah sintaks untuk hak ASSUMEROLE yang diberikan kepada pengguna dan kelompok dengan peran tertentu.

```

GRANT ASSUMEROLE
    ON { 'iam_role' [, ...] | ALL }
    TO { username | GROUP group_name | PUBLIC } [, ...]
    FOR { ALL | COPY | UNLOAD } [, ...]

```

Keterbatasan penggunaan Amazon Redshift sebagai target untuk AWS Database Migration Service

Batasan berikut berlaku saat menggunakan database Amazon Redshift sebagai target:

- Jangan aktifkan pembuatan versi untuk bucket S3 yang Anda gunakan sebagai penyimpanan perantara untuk target Amazon Redshift Anda. Jika Anda memerlukan versi S3, gunakan kebijakan siklus hidup untuk menghapus versi lama secara aktif. Jika tidak, Anda mungkin mengalami kegagalan koneksi pengujian titik akhir karena batas waktu `list-object` panggilan S3. Untuk

membuat kebijakan siklus hidup bucket S3, lihat [Mengelola siklus hidup penyimpanan](#). Untuk menghapus versi objek S3, lihat [Menghapus versi objek dari bucket berkemampuan versi](#).

- DDL berikut ini tidak didukung:

```
ALTER TABLE table name MODIFY COLUMN column name data type;
```

- AWS DMS tidak dapat bermigrasi atau mereplikasi perubahan skema dengan nama yang dimulai dengan garis bawah (_). Jika Anda memiliki skema dengan nama yang dimulai dengan garis bawah, gunakan transformasi pemetaan untuk mengubah nama skema pada target.
- Amazon Redshift tidak mendukung VARCHAR yang lebih besar dari 64 KB. LOB dari basis data tradisional tidak dapat disimpan di Amazon Redshift.
- Menerapkan pernyataan DELETE ke tabel dengan kunci primer multi-kolom tidak didukung ketika salah satu nama kolom kunci primer menggunakan kata yang dicadangkan. Pergi ke [sini](#) untuk melihat daftar kosakata cadangan pada Amazon Redshift.
- Anda mungkin mengalami masalah performa jika sistem sumber Anda melakukan operasi UPDATE pada kunci primer tabel sumber. Masalah performa ini terjadi saat menerapkan perubahan pada target. Hal ini disebabkan oleh operasi UPDATE (dan DELETE) bergantung pada nilai kunci primer untuk mengidentifikasi baris target. Jika Anda memperbarui kunci primer dari tabel sumber, log tugas Anda akan berisi pesan seperti berikut:

```
Update on table 1 changes PK to a PK that was previously updated in the same bulk update.
```

- DMS tidak mendukung nama DNS khusus saat mengonfigurasi titik akhir untuk kluster Redshift, dan Anda perlu menggunakan nama DNS yang disediakan Amazon. Karena kluster Amazon Redshift harus berada dalam akun dan wilayah AWS yang sama dengan instans replikasi, validasi gagal jika Anda menggunakan titik akhir DNS khusus.
- Amazon Redshift memiliki batas waktu sesi idle 4 jam default. Jika tidak ada aktivitas apa pun dalam tugas replikasi DMS, Redshift memutuskan sesi setelah 4 jam. Kesalahan dapat terjadi karena DMS tidak dapat terhubung dan berpotensi perlu memulai ulang. Sebagai solusinya, tetapkan batas WAKTU SESI lebih dari 4 jam untuk pengguna replikasi DMS. Atau, lihat deskripsi [ALTER USER di Panduan Pengembang](#) Database Amazon Redshift.
- Saat AWS DMS mereplikasi data tabel sumber tanpa kunci primer atau unik, latensi CDC mungkin tinggi sehingga menghasilkan tingkat kinerja yang tidak dapat diterima.

Mengonfigurasi basis data Amazon Redshift sebagai target untuk AWS Database Migration Service

AWS Database Migration Service harus dikonfigurasi untuk bekerja dengan instans Amazon Redshift. Tabel berikut menjelaskan properti konfigurasi yang tersedia untuk titik akhir Amazon Redshift.

Properti	Deskripsi
Server	Nama kluster Amazon Redshift yang Anda gunakan.
port	Nomor port untuk Amazon Redshift. Nilai default adalah 5439.
nama pengguna	Nama pengguna Amazon Redshift untuk pengguna terdaftar.
Kata sandi	Kata sandi untuk pengguna yang disebutkan di properti nama pengguna.
basis data	Nama (layanan) gudang data Amazon Redshift yang Anda gunakan.

Jika ingin menambahkan atribut string koneksi tambahan untuk titik akhir Amazon Redshift, Anda dapat menentukan atribut `maxFileSize` dan `fileTransferUploadStreams`. Untuk informasi lebih lanjut tentang atribut ini, lihat [Pengaturan titik akhir saat menggunakan Amazon Redshift sebagai target untuk AWS DMS](#).

Menggunakan perutean VPC yang disempurnakan dengan Amazon Redshift sebagai target AWS Database Migration Service

Jika Anda menggunakan Perutean VPC yang disempurnakan dengan target Amazon Redshift, semua lalu lintas COPY antara kluster Amazon Redshift dan repositori data Anda akan melewati VPC tersebut. Karena perutean VPC yang disempurnakan memengaruhi cara Amazon Redshift mengakses sumber daya lain, perintah COPY kemungkinan gagal jika Anda belum mengonfigurasi VPC Anda dengan benar.

AWS DMS dapat terpengaruh oleh perilaku ini karena menggunakan perintah COPY untuk memindahkan data di S3 menuju kluster Amazon Redshift.

Berikut ini adalah langkah yang dibutuhkan AWS DMS untuk memuat data ke sebuah target Amazon Redshift:

1. AWS DMS salin data dari sumber ke file .csv pada server replikasi.

2. AWS DMS menggunakan SDK AWS untuk menyalin file .csv ke dalam bucket S3 pada akun Anda.
3. AWS DMS kemudian menggunakan perintah COPY di Amazon Redshift untuk menyalin data dari file .csv di S3 ke tabel yang sesuai di Amazon Redshift.

Jika Perutean VPC yang disempurnakan tidak aktif, Amazon Redshift mengarahkan lalu lintas melalui internet, termasuk lalu lintas ke layanan lain dalam jaringan AWS. Jika fitur ini tidak diaktifkan Anda tidak harus mengonfigurasi jalur jaringan. Jika fitur diaktifkan, Anda harus secara khusus membuat jalur jaringan antara Anda VPC dari kluster dan sumber data Anda. Untuk informasi lebih lanjut tentang konfigurasi yang diperlukan, lihat [Perutean VPC](#) dalam dokumentasi Amazon Redshift.

Membuat dan menggunakan kunci AWS KMS untuk mengenkripsi data target Amazon Redshift

Anda dapat mengenkripsi data target yang dikirim ke Amazon S3 sebelum disalin ke Amazon Redshift. Untuk melakukannya, Anda dapat membuat dan menggunakan kunci AWS KMS khusus. Anda dapat menggunakan kunci yang Anda buat untuk mengenkripsi data target menggunakan salah satu mekanisme berikut saat Anda membuat titik akhir target Amazon Redshift:

- Gunakan pilihan berikut ketika Anda menjalankan perintah `create-endpoint` menggunakan AWS CLI.

```
--redshift-settings '{"EncryptionMode": "SSE_KMS", "ServerSideEncryptionKmsKeyId":  
  "your-kms-key-ARN"}'
```

Di sini, *your-kms-key-ARN* adalah Amazon Resource Name (ARN) untuk kunci KMS Anda. Untuk informasi lebih lanjut, lihat [Menggunakan kunci enkripsi data, dan bucket Amazon S3 sebagai penyimpanan perantara](#).

- Tetapkan atribut koneksi tambahan `encryptionMode` ke nilai `SSE_KMS` dan atribut koneksi tambahan `serverSideEncryptionKmsKeyId` ke ARN untuk kunci KMS Anda. Untuk informasi lebih lanjut lihat, [Pengaturan titik akhir saat menggunakan Amazon Redshift sebagai target untuk AWS DMS](#).

Untuk mengenkripsi data target Amazon Redshift menggunakan kunci KMS, Anda memerlukan (IAM) role AWS Identity and Access Management yang memiliki izin untuk mengakses data Amazon Redshift. IAM role ini kemudian diakses dalam sebuah kebijakan (kebijakan kunci) yang dilampirkan

ke kunci enkripsi yang Anda buat. Anda dapat melakukannya melalui konsol IAM Anda dengan membuat hal berikut:

- IAM role dengan kebijakan yang dikelola oleh AWS.
- Kunci KMS dengan kebijakan kunci yang mereferensikan peran ini.

Prosedur berikut menjelaskan cara pembuatannya.

Untuk membuat IAM role dengan kebijakan yang dikelola AWS yang diwajibkan

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Peran. Halaman Peran terbuka.
3. Pilih Buat peran. Halaman Buat peran terbuka.
4. Pilih Layanan AWS sebagai entitas terpercaya, pilih DMS sebagai layanan untuk menggunakan peran.
5. Pilih Berikutnya: Izin. Halaman Lampirkan kebijakan izin akan muncul.
6. Cari dan pilih AmazonDMSRedshiftS3Role kebijakan.
7. Pilih Berikutnya: Penandaan. Halaman Tambahkan penandaan akan muncul. Di sini, Anda dapat menambahkan tanda apa pun yang Anda inginkan.
8. Pilih Berikutnya: Tinjauan dan lihat hasil yang dikeluarkan.
9. Jika Anda membutuhkan pengaturan, masukkan nama untuk peran (misalnya, `DMS-Redshift-endpoint-access-role`), dan deskripsi tambahan apa pun, lalu pilih Buat peran. Halaman Peran terbuka dengan pesan yang menunjukkan bahwa peran Anda telah dibuat.

Anda sekarang telah membuat peran baru untuk mengakses sumber daya Amazon Redshift untuk enkripsi dengan nama tertentu, misalnya `DMS-Redshift-endpoint-access-role`.

Untuk membuat AWS KMS kunci enkripsi dengan kebijakan kunci yang mereferensikan IAM role Anda

 Note

Untuk informasi lebih lanjut tentang cara AWS DMS bekerja dengan kunci enkripsi AWS KMS, lihat [Mengatur kunci enkripsi dan menentukan izin AWS KMS](#).

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di <https://console.aws.amazon.com/kms>.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih Buat kunci. Halaman Konfigurasi kunci terbuka.
5. Untuk Tipe Kunci, pilih Simetris.

 Note

Ketika Anda membuat kunci ini, Anda hanya bisa membuat kunci simetris, karena semua layanan AWS, seperti Amazon Redshift, hanya bekerja dengan kunci enkripsi simetris.

6. Pilih Pilihan lanjutan. Untuk Asal material kunci, pastikan bahwa KMS dipilih, lalu pilih Selanjutnya. Halaman Tambahkan label terbuka.
7. Untuk Buat alias dan deskripsi, masukkan alias untuk kunci (misalnya, DMS-Redshift-endpoint-encryption-key) dan keterangan tambahan apa pun.
8. Untuk Penandaan, tambahkan tanda apa pun yang Anda inginkan untuk membantu mengidentifikasi kunci dan melacak penggunaannya, lalu pilih Selanjutnya. Halaman Tentukan izin administratif kunci terbuka dan menampilkan daftar pengguna dan peran yang dapat Anda pilih.
9. Tambahkan pengguna dan peran yang Anda inginkan untuk mengelola kunci. Pastikan bahwa pengguna dan peran ini memiliki izin yang diperlukan untuk mengelola kunci.
10. Untuk Penghapusan kunci, pilih apakah administrator kunci dapat menghapus kunci, lalu pilih Selanjutnya. Halaman Tentukan izin penggunaan kunci terbuka dan menampilkan daftar pengguna dan peran tambahan yang dapat Anda pilih.
11. Untuk Akun ini, pilih pengguna yang tersedia yang Anda inginkan untuk melakukan operasi kriptografi pada target Amazon Redshift. Juga pilih peran yang sudah Anda buat sebelumnya di Peran untuk mengaktifkan akses guna mengenkripsi objek target Amazon Redshift, misalnya DMS-Redshift-endpoint-access-role).
12. Jika Anda ingin menambahkan akun lain yang tidak terdaftar untuk memiliki akses yang sama dengan ini, pada Akun AWS Lainnya, pilih Tambahkan akun AWS lainnya, lalu pilih Selanjutnya. Halaman Meninjau dan mengedit kebijakan kunci terbuka dan menampilkan JSON untuk kebijakan kunci yang dapat Anda tinjau dan edit dengan mengetik ke JSON yang ada. Di sini, Anda dapat melihat tempat kebijakan kunci mereferensikan peran dan pengguna (misalnya, Admin dan User1) yang sudah Anda pilih pada langkah sebelumnya. Anda juga dapat melihat

tindakan kunci berbeda diizinkan untuk principals yang berbeda (pengguna dan peran), seperti yang ditunjukkan dalam contoh berikut.

```
{
  "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
        }
      }
    }
  ]
}
```

- Pilih Selesai. Halaman kunci Enkripsi terbuka dengan pesan yang menunjukkan bahwa Anda AWS KMS key telah dibuat.

Anda sekarang telah membuat bukti kunci KMS baru dengan nama alias tertentu (misalnya `DMS-Redshift-endpoint-encryption-key`). Kunci ini memungkinkan AWS DMS untuk mengenkripsi data target Amazon Redshift.

Pengaturan titik akhir saat menggunakan Amazon Redshift sebagai target untuk AWS DMS

Anda dapat menggunakan pengaturan titik akhir untuk mengonfigurasi basis data target Amazon Redshift mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir target menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--redshift-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan titik akhir yang dapat Anda gunakan dengan Amazon Redshift sebagai target.

Nama	Deskripsi
<code>MaxFileSize</code>	<p>Tentukan ukuran maksimum (dalam KB) dari setiap file .csv yang digunakan untuk mentransfer data ke Amazon Redshift.</p> <p>Nilai default: 32768 KB (32 MB)</p> <p>Nilai valid: 1-1.048.576</p> <p>Contoh: <code>--redshift-settings '{"MaxFileSize": 512}'</code></p>
<code>FileTransferUploadStreams</code>	<p>Tentukan jumlah thread yang digunakan untuk mengunggah satu file.</p> <p>Nilai default: 10</p> <p>Nilai valid: 1-64</p>

Nama	Deskripsi
	<p>Contoh: <code>--redshift-settings '{"FileTransferUploadStreams": 20}'</code></p>
Acceptanydate	<p>Tentukan apakah semua format tanggal diterima, termasuk format tanggal yang tidak valid seperti 0000-00-00. Nilai Boolean.</p> <p>Nilai default: SALAH</p> <p>Nilai valid: BETUL SALAH</p> <p>Contoh: <code>--redshift-settings '{"Acceptanydate": true}'</code></p>
Dateformat	<p>Tentukan format tanggal. Ini adalah masukan string yang secara default kosong. Format default adalah YYYY-MM-DD tetapi Anda dapat mengubahnya, misalnya menjadi DD-MM-YYYY. Jika nilai tanggal atau waktu Anda menggunakan format yang berbeda, gunakan argumen auto dengan parameter Dateformat . Argumen auto mengenali beberapa format yang tidak didukung ketika menggunakan String Dateformat . Kata kunci auto bersifat case-sensitive.</p> <p>Nilai default: kosong</p> <p>Nilai yang valid: <i>dateformat_string</i> " atau auto</p> <p>Contoh: <code>--redshift-settings '{"Dateformat": "auto}"</code></p>

Nama	Deskripsi
Timeformat	<p>Menentukan format waktu. Ini adalah masukan string yang secara default kosong. Argumen auto mengenali beberapa format yang tidak didukung ketika menggunakan string Timeformat . Jika nilai tanggal dan waktu Anda menggunakan format yang berbeda satu sama lain, gunakan argumen auto dengan parameter Timeformat .</p> <p>Nilai default: 10</p> <p>Nilai yang valid: <i>TimeFormat_String</i> "auto" "epochsecs" "epochmillisecs"</p> <p>Contoh: <code>--redshift-settings '{"Timeformat": "auto"}'</code></p>
Emptyasnull	<p>Tentukan apakah AWS DMS harus melakukan migrasi field CHAR dan VARCHAR kosong menjadi null. Nilai benar menetapkan field CHAR dan VARCHAR kosong sebagai null.</p> <p>Nilai default: SALAH</p> <p>Nilai valid: BETUL SALAH</p> <p>Contoh: <code>--redshift-settings '{"Emptyasnull": true}'</code></p>

Nama	Deskripsi
TruncateColumns	<p>Potong data dalam kolom sesuai dengan jumlah karakter sehingga cocok dengan spesifikasi kolom. Berlaku hanya pada kolom dengan tipe data VARCHAR atau CHAR, dan baris berukuran 4 MB atau kurang.</p> <p>Nilai default: SALAH</p> <p>Nilai valid: BETUL SALAH</p> <p>Contoh: <code>--redshift-settings '{"TruncateColumns": true}'</code></p>
RemoveQuotes	<p>Hapus tanda kutip yang mengelilingi string pada data yang akan masuk. Semua karakter dalam tanda kutip, termasuk delimiter, dipertahankan. Untuk informasi lebih lanjut tentang menghapus kutipan untuk target Amazon Redshift, lihat Panduan Developer Database Amazon Redshift.</p> <p>Nilai default: SALAH</p> <p>Nilai valid: BETUL SALAH</p> <p>Contoh: <code>--redshift-settings '{"RemoveQuotes": true}'</code></p>
TrimBlanks	<p>Menghapus jejak karakter spasi putih dari VARCHAR string. Parameter ini hanya berlaku untuk kolom dengan tipe data VARCHAR.</p> <p>Nilai default: SALAH</p> <p>Nilai valid: BETUL SALAH</p> <p>Contoh: <code>--redshift-settings '{"TrimBlanks": true}'</code></p>

Nama	Deskripsi
EncryptionMode	<p>Menentukan mode enkripsi sisi server yang ingin Anda gunakan untuk mengirim data Anda ke S3 sebelum disalin ke Amazon Redshift. Nilai valid adalah SSE_S3 (enkripsi sisi server S3) atau SSE_KMS (enkripsi kunci KMS). Jika Anda memilih SSE_KMS, atur parameter <code>ServerSideEncryptionKeyId</code> Untuk Amazon Resource Name (ARN) untuk kunci KMS yang akan digunakan untuk enkripsi.</p> <div data-bbox="688 640 1507 1052"><p> Note</p><p>Anda juga dapat menggunakan <code>modify-endpoint</code> perintah CLI untuk mengubah nilai <code>EncryptionMode</code> pengaturan untuk titik akhir yang ada dari ke. SSE_KMS SSE_S3 Tetapi Anda tidak dapat mengubah <code>EncryptionMode</code> nilai dari SSE_S3 ke SSE_KMS.</p></div> <p>Nilai default: SSE_S3</p> <p>Nilai valid: SSE_S3 atau SSE_KMS</p> <p>Contoh: <code>--redshift-settings '{"EncryptionMode": "SSE_S3"}</code></p>

Nama	Deskripsi
ServerSideEncryptionKmsKeyId	<p>Jika Anda mengatur EncryptionMode ke SSE_KMS, tetapkan parameter ini ke ARN untuk kunci KMS. Anda dapat menemukan ARN ini dengan memilih alias kunci dalam daftar kunci AWS KMS yang dibuat untuk akun Anda. Bila Anda membuat kunci, Anda harus mengasosiasikan kebijakan dan peran tertentu dengannya. Untuk informasi selengkapnya, lihat Membuat dan menggunakan kunci AWS KMS untuk mengenkripsi data target Amazon Redshift.</p> <p>Contoh: <code>--redshift-settings '{"ServerSideEncryptionKmsKeyId":"arn:aws:kms:us-east-1:111122223333:key/11a1a1a1-aaaa-9999-abab-2bbbbbb222a2"}'</code></p>
EnableParallelBatchInMemoryCSVFiles	<p>EnableParallelBatchInMemoryCSVFiles Pengaturan ini meningkatkan kinerja tugas beban penuh multithreaded yang lebih besar dengan meminta DMS menulis ke disk, bukan memori. Nilai bawaannya adalah false.</p>
CompressCsvFiles	<p>Gunakan atribut ini untuk mengompres data yang dikirim ke target Amazon Redshift selama migrasi. Nilai default adalah true, dan kompresi diaktifkan secara default.</p>

Menggunakan kunci enkripsi data, dan bucket Amazon S3 sebagai penyimpanan perantara

Anda dapat menggunakan Amazon Redshift target endpoint pengaturan untuk mengkonfigurasi berikut:

- Sebuah kunci enkripsi data AWS KMS kustom. Kemudian Anda dapat menggunakan kunci ini untuk mengenkripsi data Anda yang dikirim ke Amazon S3 sebelum disalin ke Amazon Redshift.
- Sebuah bucket S3 kustom sebagai penyimpanan perantara untuk data yang dimigrasi ke Amazon Redshift.

- Memetakan boolean sebagai boolean dari sumber PostgreSQL. Secara default, tipe BOOLEAN dimigrasikan sebagai varchar (1). Anda dapat menentukan `MapBooleanAsBoolean` untuk membiarkan target Redshift Anda memigrasikan jenis boolean sebagai boolean, seperti yang ditunjukkan pada contoh berikut.

```
--redshift-settings '{"MapBooleanAsBoolean": true}'
```

Perhatikan bahwa Anda harus mengatur pengaturan ini pada titik akhir sumber dan target agar dapat diterapkan.

Pengaturan kunci KMS untuk enkripsi data

Contoh berikut menunjukkan konfigurasi kunci KMS kustom untuk mengenkripsi data Anda yang dikirim ke S3. Anda bisa memulai dengan membuat panggilan `create-endpoint` menggunakan 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"}'
```

Di sini, objek JSON ditentukan oleh pilihan `--redshift-settings` yang mendefinisikan dua parameter. Satu berupa parameter `EncryptionMode` dengan nilai `SSE_KMS`. Yang satu lagi berupa parameter `ServerSideEncryptionKmsKeyId` dengan nilai `arn:aws:kms:us-east-1:111122223333:key/24c3c5a1-f34a-4519-a85b-2debbef226d1`. Nilai ini adalah Amazon Resource Name (ARN) untuk kunci KMS kustom Anda.

Secara default, enkripsi data S3 terjadi menggunakan enkripsi pada sisi server S3. Bagi target Amazon Redshift pada contoh sebelumnya, ini juga setara dengan menentukan pengaturan titik akhir, seperti dalam contoh berikut.

```
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"}'
```

Untuk informasi selengkapnya tentang bekerja dengan enkripsi sisi server S3, lihat [Melindungi data menggunakan enkripsi sisi server](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Note

Anda juga dapat menggunakan perintah CLI `modify-endpoint` untuk mengubah nilai parameter `EncryptionMode` untuk titik akhir yang ada dari `SSE_KMS` ke `SSE_S3`. Tapi Anda tidak dapat mengubah nilai `EncryptionMode` dari `SSE_S3` ke `SSE_KMS`.

Pengaturan bucket Amazon S3

Ketika Anda memigrasi data ke titik akhir target Amazon Redshift, AWS DMS menggunakan default bucket Amazon S3 sebagai penyimpanan tugas perantara sebelum menyalin data yang dimigrasi ke Amazon Redshift. Misalnya, contoh yang ditampilkan untuk membuat titik akhir target Amazon Redshift dengan kunci enkripsi AWS KMS data menggunakan bucket S3 default ini (lihat). [Pengaturan kunci KMS untuk enkripsi data](#)

Sebagai gantinya, Anda dapat menentukan bucket S3 kustom untuk penyimpanan perantara ini dengan menyertakan parameter berikut dalam nilai pilihan `--redshift-settings` Anda pada perintah AWS CLI `create-endpoint`:

- `BucketName` – String yang Anda tentukan sebagai nama penyimpanan bucket S3. Jika peran akses layanan Anda didasarkan pada `AmazonDMSRedshiftS3Role` kebijakan, nilai ini harus memiliki awalan `dms-`, misalnya, `dms-my-bucket-name`.
- `BucketFolder` – (Opsional) Sebuah string yang dapat Anda tentukan sebagai nama folder penyimpanan di bucket S3 yang sudah ditetapkan.
- `ServiceAccessRoleArn` – ARN dari IAM role yang memberikan akses administratif ke bucket S3. Umumnya, Anda membuat peran ini berdasarkan pada kebijakan `AmazonDMSRedshiftS3Role`. Sebagai contoh, lihat prosedur untuk membuat IAM role dengan kebijakan terkelola AWS di [Membuat dan menggunakan kunci AWS KMS untuk mengenkripsi data target Amazon Redshift](#).

Note

Jika Anda menetapkan ARN dari IAM role yang berbeda menggunakan pilihan `--service-access-role-arn` dari perintah `create-endpoint`, maka pilihan IAM role ini akan diutamakan.

Contoh berikut menunjukkan bagaimana Anda bisa menggunakan parameter ini untuk menentukan bucket Amazon S3 kustom dalam panggilan `create-endpoint` yang menggunakan AWS CLI berikut ini.

```
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"}'
```

Pengaturan tugas multithreaded untuk Amazon Redshift

Anda dapat meningkatkan kinerja tugas full load dan change data capture (CDC) untuk titik akhir target Amazon Redshift dengan menggunakan pengaturan tugas multithreaded. Mereka memungkinkan Anda untuk menentukan jumlah thread serentak dan jumlah catatan untuk disimpan dalam buffer.

Pengaturan tugas beban penuh multithreaded untuk Amazon Redshift

Untuk mempromosikan kinerja beban penuh, Anda dapat menggunakan pengaturan `ParallelLoad*` tugas berikut:

- `ParallelLoadThreads`— Menentukan jumlah thread bersamaan yang digunakan DMS selama pemuatan penuh untuk mendorong catatan data ke titik akhir target Amazon Redshift. Nilai default adalah nol (0) dan nilai maksimum adalah 32. Untuk informasi selengkapnya, lihat [Pengaturan tugas beban penuh](#).

Anda dapat menggunakan `enableParallelBatchInMemoryCSVFiles` atribut yang disetel `false` saat menggunakan pengaturan `ParallelLoadThreads` tugas. Atribut meningkatkan kinerja tugas beban penuh multithreaded yang lebih besar dengan meminta DMS menulis ke disk, bukan memori. Nilai bawaannya adalah `true`.

- `ParallelLoadBufferSize`— Menentukan permintaan catatan data maksimum saat menggunakan thread beban paralel dengan target Redshift. Nilai default adalah 100 dan nilai maksimum adalah 1.000. Kami menyarankan Anda menggunakan opsi ini ketika `ParallelLoadThreads > 1` (lebih besar dari satu).

Note

Support untuk penggunaan pengaturan `ParallelLoad*` tugas selama LOAD PENUH ke titik akhir target Amazon Redshift tersedia dalam AWS DMS versi 3.4.5 dan yang lebih tinggi. Pengaturan titik akhir `ReplaceInvalidChars` Redshift tidak didukung untuk digunakan selama pengambilan data perubahan (CDC) atau selama tugas migrasi BEBAN PENUH yang diaktifkan beban paralel. Hal ini didukung untuk migrasi FULL LOAD ketika beban paralel tidak diaktifkan. Untuk informasi selengkapnya lihat [RedshiftSettings](#) di Referensi AWS Database Migration Service API

Pengaturan tugas CDC multithreaded untuk Amazon Redshift

Untuk mempromosikan performa CDC, Anda dapat menggunakan pengaturan tugas `ParallelApply*` berikut:

- `ParallelApplyThreads` - Tentukan jumlah thread serentak yang digunakan AWS DMS selama beban CDC untuk mengirim data catatan ke target titik akhir Amazon Redshift. Nilai default adalah nol (0) dan nilai maksimum adalah 32. Nilai minimum yang disarankan sama dengan jumlah irisan di cluster Anda.
- `ParallelApplyBufferSize` - Tentukan permintaan catatan data maksimum saat menggunakan thread penerapan paralel dengan target Redshift. Nilai default adalah 100 dan nilai maksimum adalah 1.000. Kami merekomendasikan untuk menggunakan opsi ini ketika `ParallelApplyThreads > 1` (lebih besar dari satu).

Untuk mendapatkan manfaat yang paling besar untuk Redshift sebagai target, kami merekomendasikan bahwa nilai `ParallelApplyBufferSize` setidaknya dua kali (dobel atau lebih) dari jumlah `ParallelApplyThreads`.

Note

Support untuk penggunaan pengaturan `ParallelApply*` tugas selama titik akhir target CDC ke Amazon Redshift tersedia AWS DMS dalam versi 3.4.3 dan yang lebih tinggi.

Tingkat paralelisme yang diterapkan tergantung pada korelasi antara total Ukuran batch dan Ukuran file maksimal yang digunakan untuk mentransfer data. Ketika menggunakan pengaturan tugas CDC multithreaded dengan target Redshift, manfaat yang diperoleh ketika ukuran batch besar dalam kaitannya dengan ukuran file maksimum. Misalnya, Anda dapat menggunakan kombinasi titik akhir dan pengaturan tugas berikut untuk menyetel performa yang optimal.

```
// Redshift endpoint setting

    MaxFileSize=250000;

// Task settings

    BatchApplyEnabled=true;
    BatchSplitSize =8000;
    BatchApplyTimeoutMax =1800;
    BatchApplyTimeoutMin =1800;
    ParallelApplyThreads=32;
    ParallelApplyBufferSize=100;
```

Menggunakan pengaturan pada contoh sebelumnya, pelanggan dengan beban kerja transaksional berat mendapat manfaat dengan buffer batch mereka, berisi 8000 catatan, diisi dalam 1800 detik, memanfaatkan 32 thread paralel dengan ukuran file maksimum 250 MB.

Untuk informasi selengkapnya, lihat [Mengubah pengaturan penyetelan pemrosesan](#).

Note

DMS permintaan yang berjalan selama replikasi berkelanjutan klaster Redshift dapat berbagi sama WLM (beban kerja manajemen) antrean dengan aplikasi lain permintaan yang berjalan. Jadi, pertimbangkan untuk mengonfigurasi properti WLM dengan benar agar memengaruhi performa selama replikasi berkelanjutan ke target Redshift. Sebagai contoh, jika kueri ETL paralel lainnya berjalan, DMS berjalan lebih lambat dan peningkatan performa hilang.

Tipe data target untuk Amazon Redshift

Titik akhir Amazon Redshift untuk AWS DMS mendukung sebagian besar tipe data Amazon Redshift. Tabel berikut menampilkan tipe data target Amazon Redshift yang didukung saat menggunakan AWS DMS dan pemetaan default dari tipe data AWS DMS.

Untuk informasi tambahan tentang tipe data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Tipe data AWS DMS	Tipe data Amazon Redshift
BOOLEAN	BOOL
BYTES	VARCHAR (Length)
DATE	DATE
TIME	VARCHAR(20)
DATETIME	<p>Jika skalanya => 0 dan =< 6, tergantung pada jenis kolom target Redshift, maka salah satu dari berikut ini:</p> <p>TIMESTAMP (s)</p> <p>TIMESTAMPTZ (s) - Jika stempel waktu sumber berisi offset zona (seperti di SQL Server atau Oracle) itu dikonversi ke UTC saat disisipkan/diperbarui. Jika tidak mengandung offset, maka waktu sudah dipertimbangkan dalam UTC.</p> <p>Jika skala => 7 dan =< 9, maka:</p> <p>VARCHAR (37)</p>
INT1	INT2
INT2	INT2
INT4	INT4

Tipe data AWS DMS	Tipe data Amazon Redshift
INT8	INT8
NUMERIC	<p>Jika skala => 0 dan =< 37, maka:</p> <p>NUMERIC (p,s)</p> <p>Jika skala => 38 dan =< 127, maka:</p> <p>VARCHAR (Length)</p>
REAL4	FLOAT4
REAL8	FLOAT8
STRING	<p>Jika panjangnya 1-65.535, gunakan VARCHAR (length in bytes)</p> <p>Jika panjangnya 65.536-2.147.483.647, maka gunakan VARCHAR (65535)</p>
UINT1	INT2
UINT2	INT2
UINT4	INT4
UINT8	NUMERIC (20,0)
WSTRING	<p>Jika panjangnya 1-65.535, gunakan NVARCHAR (length in bytes)</p> <p>Jika panjangnya 65.536-2.147.483.647, gunakan NVARCHAR (65535)</p>
BLOB	<p>VARCHAR (maximum LOB size *2)</p> <p>Ukuran LOB maksimum tidak dapat melebihi 31 KB. Amazon Redshift tidak mendukung VARCHAR yang lebih besar dari 64 KB.</p>

Tipe data AWS DMS	Tipe data Amazon Redshift
NCLOB	NVARCHAR (maximum LOB size) Ukuran LOB maksimum tidak dapat melebihi 63 KB. Amazon Redshift tidak mendukung VARCHAR yang lebih besar dari 64 KB.
CLOB	VARCHAR (maximum LOB size) Ukuran LOB maksimum tidak dapat melebihi 63 KB. Amazon Redshift tidak mendukung VARCHAR yang lebih besar dari 64 KB.

Menggunakan AWS DMS Amazon Redshift Serverless sebagai Target

AWS DMS mendukung penggunaan Amazon Redshift Serverless sebagai titik akhir target. [Untuk informasi tentang menggunakan Amazon Redshift Tanpa Server, lihat Amazon Redshift Tanpa Server di Panduan Manajemen Pergeseran Merah Amazon.](#)

Topik ini menjelaskan cara menggunakan titik akhir Amazon Redshift Tanpa Server dengan AWS DMS

Note

Saat membuat titik akhir Amazon Redshift Tanpa Server, untuk DatabaseNamebidang konfigurasi [RedshiftSettings](#) titik akhir Anda, gunakan nama gudang data Amazon Redshift atau nama titik akhir grup kerja. Untuk ServerNamebidang, gunakan nilai untuk Endpoint yang ditampilkan di halaman Workgroup untuk klaster tanpa server (misalnya, default-workgroup.093291321484.us-east-1.redshift-serverless.amazonaws.com). Untuk informasi tentang membuat titik akhir, lihat [Membuat titik akhir sumber dan target](#). Untuk informasi tentang titik akhir workgroup, lihat [Menyambung ke Amazon Redshift Tanpa Server](#).

Kebijakan Kepercayaan dengan Amazon Redshift Tanpa Server sebagai target

Saat menggunakan Amazon Redshift Tanpa Server sebagai titik akhir target, Anda harus menambahkan bagian yang disorot berikut ke kebijakan kepercayaan Anda. Kebijakan kepercayaan ini melekat pada `dms-access-for-endpoint` peran.

```
{
  "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
  }
}
```

Untuk informasi selengkapnya tentang menggunakan kebijakan kepercayaan dengan AWS DMS, lihat [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#).

Keterbatasan saat menggunakan Amazon Redshift Serverless sebagai target

Menggunakan Redshift Tanpa Server sebagai target memiliki batasan sebagai berikut:

- AWS DMS hanya mendukung Amazon Redshift Serverless sebagai titik akhir di wilayah yang mendukung Amazon Redshift Tanpa Server. [Untuk informasi tentang wilayah mana yang mendukung Amazon Redshift Serverless, lihat Redshift Serverless API di titik akhir Amazon Redshift dan topik kuota di Referensi Umum. AWS](#)
- Saat menggunakan Perutean VPC yang Ditingkatkan, pastikan Anda membuat endpoint Amazon S3 di VPC yang sama dengan cluster Redshift Serverless atau Redshift Provisioned Anda. Untuk informasi selengkapnya, lihat [Menggunakan perutean VPC yang disempurnakan dengan Amazon Redshift sebagai target AWS Database Migration Service](#).
- AWS DMS Tanpa server tidak mendukung Amazon Redshift Serverless sebagai target.

Menggunakan basis data SAP ASE sebagai target untuk AWS Database Migration Service

Anda dapat memigrasikan data ke basis data SAP Adaptive Server Enterprise (ASE) —sebelumnya dikenal sebagai Sybase—menggunakan AWS DMS, dari salah satu sumber basis data manapun yang didukung.

Untuk informasi tentang versi SAP ASE yang AWS DMS mendukung sebagai target, lihat [Target untuk AWS DMS](#).

Prasyarat untuk menggunakan basis data SAP ASE sebagai target untuk AWS Database Migration Service

Sebelum Anda mulai bekerja dengan basis data SAP ASE sebagai target untuk AWS DMS, pastikan bahwa Anda memenuhi prasyarat berikut:

- Menyediakan akses akun SAP ASE ke pengguna AWS DMS. Pengguna ini harus memiliki hak baca/tulis dalam basis data SAP ASE.
- Dalam beberapa kasus, Anda dapat mereplikasi SAP ASE versi 15.7 yang diinstal pada instans Amazon EC2 di Microsoft Windows yang dikonfigurasi dengan karakter non-Latin (misalnya bahasa Cina). Dalam kasus seperti itu, AWS DMS membutuhkan SAP ASE 15.7 SP121 untuk dipasang pada mesin ASE SAP target.

Keterbatasan saat menggunakan basis data SAP ASE sebagai target untuk AWS DMS

Berikut adalah batasan yang berlaku saat menggunakan basis data SAP ASE sebagai target untuk AWS DMS:

- AWS DMS tidak mendukung tabel yang mencakup field dengan tipe data berikut. Kolom hasil replikasi dengan tipe data berikut ditampilkan sebagai null.
 - Tipe yang ditetapkan pengguna (UDT)

Pengaturan Titik Akhir saat menggunakan SAP ASE sebagai target untuk AWS DMS

Anda dapat menggunakan pengaturan titik akhir untuk mengonfigurasi basis data SAP ASE serupa dengan ketika menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir target menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--sybase-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan titik akhir yang dapat Anda gunakan SAP ASE sebagai target.

Nama	Penjelasan
Driver	<p>Tetapkan atribut ini jika Anda ingin menggunakan TLS untuk ASE versi 15.7 dan setelahnya.</p> <p>Nilai default: Adaptive Server Enterprise</p> <p>Contoh: <code>driver=Adaptive Server Enterprise 16.03.06;</code></p> <p>Nilai yang valid: Adaptive Server Enterprise 16.03.06</p>
AdditionalConnectionProperties	Parameter koneksi ODBC tambahan apapun yang ingin Anda tentukan.

Tipe data target untuk SAP ASE

Tabel berikut menampilkan tipe data target basis data SAP ASE yang didukung saat menggunakan AWS DMS dan pemetaan default dari tipe data AWS DMS.

Untuk informasi tambahan tentang tipe data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Tipe data AWS DMS	Tipe data SAP ASE
BOOLEAN	BIT
BYTES	VARBINARY (panjang)
DATE	DATE
TIME	TIME
TIMESTAMP	Jika skala => 0 dan =< 6, maka: BIGDATETIME Jika skala adalah => 7 dan =< 9, maka: VARCHAR (37)
INT1	TINYINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	NUMERIC (p,s)
REAL4	REAL
REAL8	DOUBLE PRECISION
STRING	VARCHAR (Length)
UINT1	TINYINT

Tipe data AWS DMS	Tipe data SAP ASE
UINT2	UNSIGNED SMALLINT
UINT4	UNSIGNED INTEGER
UINT8	UNSIGNED BIGINT
WSTRING	VARCHAR (Length)
BLOB	IMAGE
CLOB	UNITEXT
NCLOB	TEXT

Menggunakan Amazon S3 sebagai target untuk AWS Database Migration Service

Anda dapat memigrasi data ke Amazon S3 menggunakan AWS DMS dari salah satu sumber basis data yang didukung. Saat menggunakan Amazon S3 sebagai target dalam tugas AWS DMS, baik data penuh beban maupun data perubahan penangkapan data (CDC) ditulis ke format comma-separated value (.csv) secara default. Untuk penyimpanan yang lebih ringkas dan pilihan kueri yang lebih cepat, Anda juga memiliki pilihan untuk membuat data ditulis ke format Apache Parquet (.parquet).

AWS DMS menamai file yang dibuat selama beban penuh menggunakan penghitung heksadesimal tambahan—misalnya LOAD00001.csv, LOAD00002..., LOAD00009, LOAD0000A, dan seterusnya untuk file .csv. AWS DMS menamai file CDC menggunakan stempel waktu, misalnya 20141029-1134010000.csv. Untuk setiap tabel sumber yang berisi catatan, AWS DMS membuat folder di bawah folder target yang ditentukan (jika tabel sumber tidak kosong). AWS DMS menulis semua beban penuh dan file CDC ke bucket Amazon S3 yang ditentukan. Anda dapat mengontrol ukuran file yang AWS DMS dibuat dengan menggunakan pengaturan [MaxFileSize](#) endpoint.

Parameter `bucketFolder` berisi lokasi di mana file .csv atau .parquet disimpan sebelum diunggah ke bucket S3. Dengan file .csv, data tabel disimpan dalam format berikut dalam bucket S3, ditampilkan dengan file beban penuh.

```
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
...
```

Anda dapat menentukan delimiter kolom, delimiter baru, dan parameter lainnya menggunakan atribut koneksi tambahan. Untuk informasi lebih lanjut tentang atribut koneksi tambahan, lihat [Pengaturan titik akhir saat menggunakan Amazon S3 sebagai target untuk AWS DMS](#) pada akhir bagian ini.

Anda dapat menentukan pemilik bucket dan mencegah sniping dengan menggunakan setelan endpoint `ExpectedBucketOwner` Amazon S3, seperti yang ditunjukkan berikut ini. Kemudian, saat Anda membuat permintaan untuk menguji koneksi atau melakukan migrasi, S3 memeriksa ID akun pemilik bucket terhadap parameter yang ditentukan.

```
--s3-settings='{"ExpectedBucketOwner": "AWS_Account_ID"}
```

Saat Anda menggunakan AWS DMS untuk mereplikasi perubahan data menggunakan tugas CDC, kolom pertama dari file keluaran `.csv` atau `.parquet` menunjukkan bagaimana data baris diubah seperti yang diperlihatkan untuk file `.csv` berikut.

```
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
```

Untuk contoh ini, anggaplah ada tabel `EMPLOYEE` di basis data sumber. AWS DMS menulis data ke file `.csv` atau `.parquet`, sebagai respons terhadap peristiwa berikut:

- Seorang karyawan baru (Bob Smith, ID karyawan 101) dipekerjakan pada 4-Jun-14 di kantor New York. Dalam file `.csv` atau `.parquet`, I di kolom pertama menunjukkan bahwa baris baru diINSERTkan ke tabel `EMPLOYEE` di basis data sumber.
- Pada 8-Okt-15, Bob pindah ke kantor Los Angeles. Dalam file `.csv` atau `.parquet`, U menunjukkan bahwa baris yang sesuai dalam tabel `EMPLOYEE` diUPDATE untuk mencerminkan lokasi kantor

Bob yang baru. Sisa barisan mencerminkan baris dalam tabel EMPLOYEE seperti yang muncul setelah UPDATE.

- Pada 13-Mar-17, Bob ditransfer lagi ke kantor Dallas. Dalam file .csv atau .parquet, U menunjukkan bahwa baris ini diUPDATE lagi. Sisa barisan mencerminkan baris dalam tabel EMPLOYEE seperti yang muncul setelah UPDATE.
- Setelah beberapa waktu bekerja di Dallas, Bob meninggalkan perusahaan. Dalam file .csv atau .parquet, D menunjukkan bahwa baris ini diDELETEDkan di tabel sumber. Sisa barisan mencerminkan bagaimana tampilan baris dalam tabel EMPLOYEE sebelum data baris dihapus.

Perhatikan bahwa secara default untuk CDC, AWS DMS menyimpan perubahan baris untuk setiap tabel basis data tanpa memperhatikan urutan transaksi. Jika Anda ingin menyimpan perubahan baris dalam file CDC sesuai urutan transaksi, Anda harus menggunakan pengaturan titik akhir S3 untuk menentukan hal ini dan jalur folder di mana Anda ingin file transaksi CDC disimpan pada target S3. Untuk informasi lebih lanjut, lihat [Perubahan pengambilan data \(CDC\) termasuk transaksi order pada target S3](#).

Untuk mengontrol frekuensi menulis ke target Amazon S3 selama tugas replikasi data, Anda dapat mengonfigurasi atribut koneksi tambahan `cdcMaxBatchInterval` dan `cdcMinFileSize`. Hal ini dapat menghasilkan performa yang lebih baik ketika menganalisis data tanpa operasi overhead tambahan. Untuk informasi lebih lanjut, lihat [Pengaturan titik akhir saat menggunakan Amazon S3 sebagai target untuk AWS DMS](#)

Topik

- [Prasyarat penggunaan Amazon S3 sebagai target](#)
- [Keterbatasan penggunaan Amazon S3 sebagai target](#)
- [Keamanan](#)
- [Menggunakan Apache Parquet untuk menyimpan objek Amazon S3](#)
- [Penandaan objek Amazon S3](#)
- [Membuat kunci AWS KMS untuk mengenkripsi objek target Amazon S3](#)
- [Menggunakan partisi folder berdasarkan tanggal](#)
- [Beban paralel sumber yang dipartisi saat menggunakan Amazon S3 sebagai target AWS DMS](#)
- [Pengaturan titik akhir saat menggunakan Amazon S3 sebagai target untuk AWS DMS](#)
- [Menggunakan AWS Glue Data Catalog dengan target Amazon S3 untuk AWS DMS](#)
- [Menggunakan enkripsi data, file parket, dan CDC pada target Amazon S3 Anda](#)

- [Menunjukkan operasi sumber DB dalam data S3 bermigrasi](#)
- [Tipe data target untuk S3 Parquet](#)

Prasyarat penggunaan Amazon S3 sebagai target

Sebelum menggunakan Amazon S3 sebagai target, pastikan bahwa hal berikut ini benar:

- Bucket S3 yang Anda gunakan sebagai target berada dalam wilayah AWS yang sama dengan instans replikasi DMS yang Anda gunakan untuk memigrasi data.
- Akun AWS yang Anda gunakan untuk migrasi memiliki IAM role dengan akses menulis dan menghapus ke bucket S3 yang Anda gunakan sebagai target.
- Peran ini memiliki akses penandaan sehingga Anda dapat menandai objek S3 apa pun yang ditulis ke target bucket.
- IAM role memiliki DMS (dms.amazonaws.com) yang ditambahkan sebagai entitas terpercaya.

Untuk mengatur akses akun ini, pastikan peran yang diberikan ke akun pengguna yang digunakan untuk membuat tugas migrasi memiliki serangkaian izin berikut.

```
{
  "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"
      ]
    }
  ]
}
```

```
    ]
  }
]
}
```

Untuk prasyarat untuk menggunakan validasi dengan S3 sebagai target, lihat [Prasyarat validasi target S3](#)

Keterbatasan penggunaan Amazon S3 sebagai target

Batasan berikut berlaku saat menggunakan Amazon S3 sebagai target:

- Jangan aktifkan pembuatan versi untuk S3. Jika Anda memerlukan versi S3, gunakan kebijakan siklus hidup untuk menghapus versi lama secara aktif. Jika tidak, Anda mungkin mengalami kegagalan koneksi pengujian titik akhir karena batas waktu `list-object` panggilan S3. Untuk membuat kebijakan siklus hidup bucket S3, lihat [Mengelola siklus hidup](#) penyimpanan. Untuk menghapus versi objek S3, lihat [Menghapus versi objek dari bucket berkemampuan versi](#).
- Bucket S3 berkemampuan VPC (gateway VPC) didukung dalam versi 3.4.7 dan yang lebih tinggi.
- Perintah bahasa definisi data berikut (DDL) didukung untuk perubahan penangkapan data (CDC): Truncate Table, Drop Table, Create Table, Rename Table, Add Column, Drop Column, Rename Column, dan Change COLUMN Data Type. Perhatikan bahwa ketika kolom ditambahkan, dihapus, atau diganti namanya pada database sumber, tidak ada pernyataan ALTER yang direkam dalam bucket S3 target, dan AWS DMS tidak mengubah catatan yang dibuat sebelumnya agar sesuai dengan struktur baru. Setelah perubahan, AWS DMS membuat catatan baru menggunakan struktur tabel baru.

Note

Operasi pemotongan DDL menghapus semua file dan folder tabel yang terkait dari bucket S3. Anda dapat menggunakan pengaturan tugas untuk menonaktifkan perilaku itu dan mengonfigurasi cara DMS menangani perilaku DDL selama pengambilan perubahan data (CDC). Untuk informasi lebih lanjut, lihat [Pengaturan tugas untuk perubahan pengolahan DDL penanganan](#).

- Mode LOB penuh tidak didukung.
- Perubahan pada struktur tabel sumber selama beban penuh tidak didukung. Perubahan data didukung selama beban penuh.

- Beberapa tugas yang mereplikasi data dari tabel sumber yang sama untuk titik akhir bucket S3 target yang sama, mengakibatkan tugas-tugas itu menulis ke dalam file yang sama. Kami sarankan Anda menentukan titik akhir target yang berbeda (buckets) jika sumber data Anda berasal dari tabel yang sama.
- BatchApply tidak didukung untuk titik akhir S3. Menggunakan penerapan Batch (misalnya, pengaturan tugas metadata target BatchApplyEnabled) untuk target S3 dapat mengakibatkan hilangnya data.
- Anda tidak dapat menggunakan DatePartitionEnabled atau addColumnName bersama-sama dengan PreserveTransactions atau CdcPath.
- AWS DMS tidak mendukung penggantian nama beberapa tabel sumber ke folder target yang sama menggunakan aturan transformasi.
- Jika ada penulisan intensif ke tabel sumber selama fase beban penuh, DMS dapat menulis catatan duplikat ke bucket S3 atau perubahan cache.
- Jika Anda mengonfigurasi tugas dengan TargetTablePrepMode of DO_NOTHING, DMS dapat menulis catatan duplikat ke bucket S3 jika tugas berhenti dan dilanjutkan secara tiba-tiba selama fase pemuatan penuh.
- Jika Anda mengonfigurasi titik akhir target dengan PreserveTransactions pengaturan true, memuat ulang tabel tidak akan menghapus file CDC yang dihasilkan sebelumnya. Untuk informasi selengkapnya, lihat [Perubahan pengambilan data \(CDC\) termasuk transaksi order pada target S3](#).

Untuk batasan penggunaan validasi dengan S3 sebagai target, lihat [Batasan untuk menggunakan validasi target S3](#)

Keamanan

Untuk menggunakan Amazon S3 sebagai target, akun yang digunakan untuk migrasi harus memiliki akses tulis dan hapus terhadap bucket Amazon S3 yang digunakan sebagai target. Tentukan Amazon Resource Name (ARN) dari IAM role yang memiliki izin yang diperlukan untuk mengakses Amazon S3.

AWS DMS mendukung serangkaian pemberian yang telah ditentukan sebelumnya untuk Amazon S3, yang dikenal sebagai daftar kontrol akses (ACL) terekam. Setiap ACL terekam memiliki serangkaian penerima dan izin yang dapat Anda gunakan untuk mengatur izin bucket Amazon S3. Anda dapat menentukan ACL terekam menggunakan cannedACLForObjects pada atribut string koneksi untuk titik akhir target S3 Anda. Untuk informasi lebih lanjut tentang penggunaan atribut koneksi tambahan

cannedACLForObjects, lihat [Pengaturan titik akhir saat menggunakan Amazon S3 sebagai target untuk AWS DMS](#). Untuk informasi lebih lanjut tentang ACL terekam Amazon S3, lihat [ACL Terekam](#).

IAM role yang Anda gunakan untuk migrasi harus dapat melakukan Operasi API `s3:PutObjectACL`.

Menggunakan Apache Parquet untuk menyimpan objek Amazon S3

Format comma-separated value (.csv) adalah format penyimpanan default untuk objek target Amazon S3. Untuk penyimpanan yang lebih ringkas dan kueri lebih cepat, sebagai gantinya Anda bisa menggunakan Apache Parquet (.parquet) sebagai format penyimpanan.

Apache Parquet adalah format penyimpanan file sumber terbuka yang awalnya dirancang untuk Hadoop. Untuk informasi lebih lanjut tentang Apache Parquet, lihat <https://parquet.apache.org/>.

Untuk mengatur .parquet sebagai format penyimpanan untuk objek target S3 dimigrasikan, Anda dapat menggunakan mekanisme berikut:

- Pengaturan titik akhir yang Anda berikan sebagai parameter objek JSON ketika Anda membuat titik akhir menggunakan AWS CLI atau API untuk AWS DMS. Untuk informasi lebih lanjut, lihat [Menggunakan enkripsi data, file parquet, dan CDC pada target Amazon S3 Anda](#).
- Atribut koneksi tambahan yang Anda berikan sebagai daftar yang dipisahkan titik koma saat Anda membuat titik akhir. Untuk informasi lebih lanjut, lihat [Pengaturan titik akhir saat menggunakan Amazon S3 sebagai target untuk AWS DMS](#).

Penandaan objek Amazon S3

Anda dapat menandai objek Amazon S3 yang dibuat oleh instans replikasi dengan menentukan objek JSON yang sesuai sebagai bagian dari aturan pemetaan tabel tugas. Untuk informasi selengkapnya tentang persyaratan dan opsi untuk penandaan objek S3, termasuk nama tag yang valid, lihat [Penandaan objek](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. Untuk informasi lebih lanjut tentang pemetaan tabel menggunakan JSON, lihat [Menentukan pemilihan tabel dan transformasi aturan menggunakan JSON](#).

Anda menandai objek S3 yang dibuat untuk tabel dan skema tertentu dengan menggunakan satu atau lebih objek JSON dari jenis aturan `selection`. Anda kemudian mengikuti objek `selection` ini (atau beberapa objek) oleh satu atau lebih objek JSON dari jenis aturan `post-processing` dengan tindakan `add-tag`. Aturan pasca-pemrosesan ini mengidentifikasi objek S3 yang ingin Anda tandai dan menentukan nama dan nilai-nilai tanda yang ingin Anda tambahkan ke objek S3 ini.

Anda dapat menemukan parameter yang akan ditetapkan dalam objek JSON dari tipe aturan `post-processing` dalam tabel berikut.

Parameter	Kemungkinan nilai	Deskripsi
<code>rule-type</code>	<code>post-processing</code>	Nilai yang menerapkan tindakan pasca pemrosesan untuk objek target yang dihasilkan. Anda dapat menentukan satu atau lebih aturan pasca-pemrosesan untuk menandai objek S3 yang dipilih.
<code>rule-id</code>	Nilai numerik.	Nilai numerik yang unik untuk mengidentifikasi aturan.
<code>rule-name</code>	Nilai alfanumerik.	Nama unik untuk mengidentifikasi aturan.
<code>rule-action</code>	<code>add-tag</code>	Tindakan pasca-pemrosesan yang ingin Anda terapkan ke objek S3. Anda dapat menambahkan satu atau lebih tanda menggunakan satu objek pasca-pemrosesan JSON tindakan <code>add-tag</code> .
<code>object-locator</code>	<p><code>schema-name</code> – Nama skema tabel.</p> <p><code>table-name</code> – Nama tabel.</p>	<p>Nama setiap skema dan tabel tempat aturannya berlaku. Anda dapat menggunakan tanda “%” persen sebagai wildcard untuk semua atau sebagian dari nilai masing-masing parameter <code>object-locator</code> . Dengan demikian, Anda bisa mencocokkan item berikut:</p> <ul style="list-style-type: none"> • Sebuah tabel tunggal dalam skema tunggal • Sebuah tabel tunggal di beberapa atau semua skema

Parameter	Kemungkinan nilai	Deskripsi
		<ul style="list-style-type: none">• Beberapa atau semua tabel dalam skema tunggal• Beberapa atau semua tabel di beberapa atau semua skema

Parameter	Kemungkinan nilai	Deskripsi
tag-set	<p>key – Nama apa pun yang valid untuk satu tanda.</p> <p>value – Nilai JSON apa pun yang valid untuk tanda ini.</p>	<p>Nama dan nilai-nilai untuk satu atau beberapa tanda yang ingin Anda tetapkan pada setiap objek S3 dibuat cocok dengan yang ditentukan <code>object-locator</code>. Anda dapat menentukan hingga 10 pasangan nilai-kunci dalam satu objek parameter tag-set. Untuk informasi selengkapnya tentang penandaan objek S3, lihat Penandaan objek di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.</p> <p>Anda juga dapat menentukan nilai dinamis untuk semua atau sebagian dari nilai untuk parameter key dan value dari tanda menggunakan <code>\${dyn-value}</code>. Di sini, <code>\${dyn-value}</code> dapat berupa <code>\${schema-name}</code> maupun <code>\${table-name}</code>. Sehingga Anda dapat memasukkan nama skema atau tabel yang sedang dipilih atau tabel secara keseluruhan atau bagian mana pun dari nilai parameter.</p> <div data-bbox="976 1482 1507 1873" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <div data-bbox="1052 1598 1479 1873" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important</p> <p>Jika Anda memasukkan nilai dinamis untuk parameter key, Anda</p> </div> </div>

Parameter	Kemungkinan nilai	Deskripsi
		<p>dapat menghasilkan tanda dengan nama duplikat untuk objek S3, tergantung bagaimana Anda menggunakannya. Dalam kasus ini, hanya satu dari pengaturan tanda duplikat ditambahkan ke objek.</p>

Ketika Anda menentukan beberapa jenis aturan post-processing untuk menandai pilihan objek S3, setiap objek S3 ditandai menggunakan hanya satu objek tag-set dari satu aturan pasca-pemrosesan. Seperangkat tanda tertentu yang digunakan untuk menandai objek S3 yang diberikan adalah yang berasal dari aturan pasca-pemrosesan yang pencari objek terkaitnya paling sesuai dengan objek S3 tersebut.

Misalnya anggaplah bahwa dua aturan pasca-pemrosesan mengidentifikasi objek S3 yang sama. Misalkan juga pencari objek dari satu aturan menggunakan wildcard dan pencari objek dari aturan lain menggunakan exact match untuk mengidentifikasi objek S3 (tanpa wildcard). Dalam hal ini, seperangkat tanda yang terkait dengan aturan pasca-pemrosesan dengan exact match digunakan untuk menandai objek S3. Jika beberapa aturan pasca-pemrosesan cocok dengan objek S3 yang diberikan dengan sama baiknya, seperangkat tanda yang terkait dengan aturan pasca-pemrosesan pertama digunakan untuk menandai objek.

Example Menambahkan tanda statis ke objek S3 yang dibuat untuk tabel dan skema tunggal

Aturan pemilihan dan pasca-pemrosesan berikut menambahkan tiga tanda (tag_1, tag_2, dan tag_3 dengan nilai statis yang sesuai value_1, value_2, dan value_3) ke objek S3 yang sudah dibuat. Objek S3 ini berkaitan dengan sebuah tabel di sumber bernama STOCK dengan skema bernama 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 Menambahkan tanda statis ke objek S3 yang dibuat untuk beberapa tabel dan skema

Contoh berikut memiliki satu pilihan dan dua aturan pasca-pemrosesan, dengan masukan dari sumber mencakup semua tabel dan semua skema mereka.

```
{
  "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"
      }
    ]
  }
]
}

```

Aturan pasca-pemrosesan pertama menambahkan dua tanda (`dw-schema-name` dan `dw-schema-table`) dengan nilai dinamis yang sesuai (`${schema-name}` dan `my_prefix_${table-name}`) untuk hampir semua objek S3 yang dibuat dalam target. Kecuali ketika objek S3 diidentifikasi dan ditandai dengan aturan pasca-pemrosesan kedua. Sehingga setiap objek target S3 yang teridentifikasi oleh pencari objek wildcard dibuat dengan penanda yang mengidentifikasi skema dan tabel yang sesuai dengan sumbernya.

Aturan pasca-pemrosesan kedua menambahkan `tag_1` dan `tag_2` dengan nilai statis yang sesuai dengan `value_1` dan `value_2` ke objek S3 yang sudah dibuat yang teridentifikasi oleh pencari objek yang sama persis. Objek S3 yang dibuat ini sesuai dengan tabel tunggal di sumber bernama `ITEM` dengan skema bernama `aat`. Karena objek yang sama persis, tanda ini menggantikan tanda apa pun pada objek yang ditambahkan dari aturan pasca-pemrosesan pertama ini, yang membandingkan objek S3 hanya berdasarkan wildcard saja.

Example Menambahkan nilai maupun nama tanda dinamis ke objek S3

Contoh berikut memiliki dua aturan seleksi dan satu aturan pasca-pemrosesan. Di sini, masukan dari sumber hanya menyertakan tabel `ITEM` di skema `retail` atau `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"
        }
      ]
    }
  ]
}

```

Tanda yang ditetapkan untuk aturan pasca-pemrosesan menambahkan dua tanda (`dw-schema-name` dan `dw-schema-table`) untuk semua objek S3 yang dibuat untuk tabel ITEM di target. Tanda pertama mempunyai nilai dinamis `"${schema-name}"` dan tanda kedua memiliki nilai statis,

"my_prefix_ITEM". Jadi, setiap objek target S3 dibuat dengan tanda yang mengidentifikasi skema dan tabel yang sesuai dengan sumbernya.

Selain itu, sekelompok tanda menambahkan dua tanda tambahan dengan nama dinamis (`${schema-name}_ITEM_tag_1` dan `"${schema-name}_ITEM_tag_2"`). Tanda ini memiliki nilai statis yang sesuai `value_1` dan `value_2`. Sehingga penanda ini masing-masing dinamai untuk skema saat ini, `retail` atau `wholesale`. Anda tidak dapat membuat salinan penanda nama dinamis dalam objek ini karena setiap objek dibuat untuk satu nama skema unik. Nama skema digunakan untuk membuat nama tanda yang unik.

Membuat kunci AWS KMS untuk mengenkripsi objek target Amazon S3

Anda dapat membuat dan menggunakan kunci AWS KMS khusus untuk mengenkripsi objek target Amazon S3 Anda. Setelah membuat kunci KMS, Anda dapat menggunakannya untuk mengenkripsi objek menggunakan salah satu pendekatan berikut ketika Anda membuat titik akhir target S3:

- Gunakan pilihan berikut untuk objek target S3 (dengan format penyimpanan file `.csv` default) ketika Anda menjalankan perintah `create-endpoint` menggunakan 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"}'
```

Di sini, *your-KMS-key-ARN* adalah Amazon Resource Name (ARN) untuk kunci KMS Anda. Untuk informasi selengkapnya, lihat [Menggunakan enkripsi data, file parquet, dan CDC pada target Amazon S3 Anda](#).

- Tetapkan atribut koneksi tambahan `encryptionMode` ke nilai `SSE_KMS` dan atribut koneksi tambahan `serverSideEncryptionKmsKeyId` ke ARN untuk kunci KMS Anda. Untuk informasi selengkapnya, lihat [Pengaturan titik akhir saat menggunakan Amazon S3 sebagai target untuk AWS DMS](#).

Untuk mengenkripsi objek target Amazon S3 menggunakan kunci KMS, Anda memerlukan peran IAM yang memiliki izin untuk mengakses bucket Amazon S3. IAM role ini kemudian diakses dalam sebuah kebijakan (kebijakan kunci) yang dilampirkan ke kunci enkripsi yang Anda buat. Anda dapat melakukannya melalui konsol IAM Anda dengan membuat hal berikut:

- Kebijakan dengan izin untuk mengakses bucket Amazon S3.

- IAM role dengan kebijakan ini.
- Kunci enkripsi kunci KMS dengan kebijakan kunci yang mereferensikan peran ini.

Prosedur berikut menjelaskan cara pembuatannya.

Membuat kebijakan IAM dengan izin mengakses bucket Amazon S3

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Kebijakan. Halaman Kebijakan terbuka.
3. Pilih Buat kebijakan. Halaman Buat kebijakan terbuka.
4. Pilih Layanan dan pilih S3. Daftar izin tindakan ditampilkan.
5. Pilih Perluas semua untuk memperluas daftar dan minimal pilih izin berikut:
 - ListBucket
 - PutObject
 - DeleteObject

Pilih izin lain yang Anda butuhkan, lalu pilih Kolaps semua untuk menutup daftar.

6. Pilih Sumber Daya Untuk menentukan sumber daya yang ingin Anda akses. Minimal, pilih Semua sumber daya Untuk menyediakan akses sumber daya Amazon S3 secara umum.
7. Tambahkan kondisi atau izin lain apa pun yang Anda butuhkan, lalu pilih Tinjau kebijakan. Periksa hasil Anda di halaman Tinjau kebijakan.
8. Jika pengaturannya sudah sesuai kebutuhan Anda, masukkan nama untuk kebijakan (misalnya DMS-S3-endpoint-access), dan berikan deskripsi apa pun, lalu pilih Buat kebijakan. Halaman Kebijakan terbuka dengan pesan yang menunjukkan bahwa kebijakan Anda telah dibuat.
9. Cari dan pilih nama kebijakan di daftar Kebijakan. Halaman Ringkasan muncul dengan menampilkan JSON untuk kebijakan yang serupa dengan berikut ini.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
```

```
        "s3:ListBucket",
        "s3:DeleteObject"
    ],
    "Resource": "*"
}
]
```

Anda sekarang telah membuat kebijakan baru untuk mengakses sumber daya Amazon S3 untuk enkripsi dengan nama tertentu, misalnya `DMS-S3-endpoint-access`.

Membuat IAM role menggunakan kebijakan ini

1. Di panel navigasi konsol IAM, pilih Peran. Halaman detail Peran terbuka.
2. Pilih Buat peran. Halaman Buat peran terbuka.
3. Dengan layanan AWS sudah dipilih sebagai entitas terpercaya, pilih DMS sebagai layanan untuk menggunakan IAM role.
4. Pilih Berikutnya: Izin. Tampilan Lampirkan kebijakan izin muncul di halaman Buat peran.
5. Temukan dan pilih kebijakan IAM untuk IAM role yang Anda buat di prosedur sebelumnya (`DMS-S3-endpoint-access`).
6. Pilih Berikutnya: Penandaan. Tampilan Tambahkan penandaan muncul di halaman Buat peran. Di sini, Anda dapat menambahkan tanda apa pun yang Anda inginkan.
7. Pilih Berikutnya: Tinjauan. Tampilan Ulasan muncul di halaman Buat peran. Di sini Anda dapat memverifikasi hasilnya.
8. Jika pengaturan sudah sesuai dengan kebutuhan Anda, masukkan nama untuk peran (wajib, misalnya, `DMS-S3-endpoint-access-role`), dan deskripsi tambahan apa pun, lalu pilih Buat peran. Halaman detail Peran terbuka dengan pesan yang menunjukkan bahwa peran Anda telah dibuat.

Anda sekarang telah membuat peran baru untuk mengakses sumber daya Amazon S3 untuk enkripsi dengan nama tertentu, misalnya, `DMS-S3-endpoint-access-role`.

Untuk membuat kunci enkripsi kunci KMS dengan kebijakan kunci yang mereferensikan peran IAM Anda

 Note

Untuk informasi lebih lanjut tentang cara AWS DMS bekerja dengan kunci enkripsi AWS KMS, lihat [Mengatur kunci enkripsi dan menentukan izin AWS KMS](#).

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di <https://console.aws.amazon.com/kms>.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih Buat kunci. Halaman Konfigurasi kunci terbuka.
5. Untuk Tipe Kunci, pilih Simetris.

 Note

Ketika Anda membuat kunci ini, Anda hanya dapat membuat kunci simetris, karena semua layanan AWS, seperti Amazon S3, hanya bekerja dengan kunci enkripsi simetris.

6. Pilih Pilihan lanjutan. Untuk Asal material kunci, pastikan bahwa KMS dipilih, lalu pilih Selanjutnya. Halaman Tambahkan label terbuka.
7. Untuk Buat alias dan deskripsi, masukkan alias untuk kunci (misalnya, DMS-S3-endpoint-encryption-key) dan keterangan tambahan apa pun.
8. Untuk Penandaan, tambahkan tanda apa pun yang Anda inginkan untuk membantu mengidentifikasi kunci dan melacak penggunaannya, lalu pilih Selanjutnya. Halaman Tentukan izin administratif kunci terbuka dan menampilkan daftar pengguna dan peran yang dapat Anda pilih.
9. Tambahkan pengguna dan peran yang Anda inginkan untuk mengelola kunci. Pastikan bahwa pengguna dan peran ini memiliki izin yang diperlukan untuk mengelola kunci.
10. Untuk Penghapusan kunci, pilih apakah administrator kunci dapat menghapus kunci, lalu pilih Selanjutnya. Halaman Tentukan izin penggunaan kunci terbuka dengan menampilkan daftar pengguna dan peran tambahan yang dapat Anda pilih.
11. Untuk Akun ini, pilih pengguna yang tersedia Anda inginkan untuk melakukan operasi kriptografi pada target Amazon S3. Pilih juga peran yang sebelumnya Anda buat di Peran

untuk mengaktifkan akses untuk mengenkripsi objek target Amazon S3, misalnya `DMS-S3-endpoint-access-role`).

12. Jika Anda ingin menambahkan akun lain yang tidak terdaftar untuk memiliki akses yang sama dengan ini, pada Akun AWS lainnya, pilih Tambahkan akun AWS lainnya, lalu pilih Selanjutnya. Halaman Meninjau dan mengedit kebijakan kunci terbuka dan menampilkan JSON untuk kebijakan kunci yang dapat Anda tinjau dan edit dengan mengetik ke JSON yang ada. Di sini, Anda dapat melihat tempat kebijakan kunci mereferensikan peran dan pengguna (misalnya, Admin dan User1) yang sudah Anda pilih pada langkah sebelumnya. Anda juga dapat melihat tindakan kunci berbeda yang diizinkan untuk principal yang berbeda (pengguna dan peran), seperti yang ditunjukkan pada contoh berikut.

```
{
  "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. Pilih Selesai. Halaman kunci Enkripsi terbuka dengan pesan yang menunjukkan bahwa kunci KMS Anda telah dibuat.

Anda sekarang telah membuat bukti kunci KMS baru dengan nama alias tertentu (misalnya `DMS-S3-endpoint-encryption-key`). Kunci ini memungkinkan AWS DMS untuk mengenkripsi objek target Amazon S3.

Menggunakan partisi folder berdasarkan tanggal

AWS DMS mendukung partisi folder S3 berdasarkan tanggal melakukan transaksi saat Anda menggunakan Amazon S3 sebagai titik akhir target Anda. Dengan menggunakan partisi folder berdasarkan tanggal, Anda dapat menulis data dari tabel sumber tunggal ke struktur folder dengan hirarki waktu dalam bucket S3. Dengan partisi folder saat membuat titik akhir target S3, Anda dapat melakukan hal berikut:

- Mengelola objek S3 Anda dengan lebih baik
- Membatasi ukuran setiap folder S3
- Mengoptimalkan kueri danau data atau operasi lain setelahnya

Anda dapat mengaktifkan partisi folder berdasarkan tanggal ketika Anda membuat titik akhir S3. Anda dapat mengaktifkannya ketika memigrasikan data yang ada dan mereplikasi perubahan yang sedang berlangsung (beban penuh + CDC), atau hanya mereplikasi perubahan data (CDC saja). Gunakan pengaturan titik akhir target berikut:

- `DatePartitionEnabled` – Menentukan partisi berdasarkan tanggal. Mengatur pilihan Boolean ini ke `true` untuk partisi folder bucket S3 berdasarkan tanggal melakukan transaksi.

Anda tidak dapat menggunakan pengaturan ini dengan `PreserveTransactions` atau `CdcPath`.

Nilai default adalah `false`.

- `DatePartitionSequence` – Mengidentifikasi urutan format tanggal yang digunakan selama partisi folder. Tetapkan pilihan ENUM ini ke `YYYYMMDD`, `YYYYMMDDHH`, `YYYYMM`, `MMYYYYDD`, atau `DDMMYYYY`. Nilai default adalah `YYYYMMDD`. Gunakan pengaturan ini saat `DatePartitionEnabled` diatur ke `true`.
- `DatePartitionDelimiter` – Menentukan pembatas pemisahan tanggal untuk digunakan selama partisi folder. Tetapkan pilihan ENUM ini ke `SLASH`, `DASH`, `UNDERSCORE`, atau `NONE`. Nilai default adalah `SLASH`. Gunakan pengaturan ini saat `DatePartitionEnabled` diatur ke `true`.

Contoh berikut menunjukkan cara mengaktifkan partisi folder berdasarkan tanggal, dengan nilai default untuk urutan partisi data dan pembatas. Dengan menggunakan pilihan `--s3-settings '{json-settings}'` dari perintah `AWS CLI.create-endpoint`.

```
--s3-settings '{"DatePartitionEnabled": true,"DatePartitionSequence":  
"YYYYMMDD","DatePartitionDelimiter": "SLASH"}
```

Beban paralel sumber yang dipartisi saat menggunakan Amazon S3 sebagai target AWS DMS

Anda dapat mengonfigurasi beban penuh paralel sumber data yang dipartisi ke target Amazon S3. Pendekatan ini meningkatkan waktu muat untuk memigrasikan data yang dipartisi dari mesin basis data sumber yang didukung ke target S3. Untuk meningkatkan waktu muat data sumber yang dipartisi, Anda membuat subfolder target S3 yang dipetakan ke partisi setiap tabel dalam database sumber. Subfolder terikat partisi ini memungkinkan untuk AWS DMS menjalankan proses paralel untuk mengisi setiap subfolder pada target.

Untuk mengonfigurasi beban penuh paralel dari target S3, S3 mendukung tiga tipe `parallel-load` aturan untuk `table-settings` aturan pemetaan tabel:

- `partitions-auto`
- `partitions-list`
- `ranges`

Untuk informasi selengkapnya tentang jenis aturan beban paralel ini, lihat [Tabel dan koleksi pengaturan aturan dan operasi](#)

Untuk tipe `partitions-auto` dan `partitions-list` aturan, AWS DMS gunakan setiap nama partisi dari titik akhir sumber untuk mengidentifikasi struktur subfolder target, sebagai berikut.

```
bucket_name/bucket_folder/database_schema_name/table_name/partition_name/
LOADseq_num.csv
```

Di sini, jalur subfolder tempat data dimigrasikan dan disimpan pada target S3 menyertakan `partition_name` subfolder tambahan yang sesuai dengan partisi sumber dengan nama yang sama. `partition_name` Subfolder ini kemudian menyimpan satu atau lebih `LOADseq_num.csv` file yang berisi data yang dimigrasikan dari partisi sumber yang ditentukan. Di sini, `seq_num` adalah postfix nomor urut pada nama file.csv, seperti `00000001` di file.csv dengan nama, `LOAD00000001.csv`

Namun, beberapa mesin database, seperti MongoDB dan DocumentDB, tidak memiliki konsep partisi. Untuk mesin database ini, AWS DMS tambahkan indeks segmen sumber berjalan sebagai awalan ke nama file.csv target, sebagai berikut.

```
.../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
```

Di sini, file `SEGMENT1_LOAD00000001.csv` dan `SEGMENT1_LOAD00000002.csv` diberi nama dengan awalan indeks segmen sumber berjalan yang sama, `SEGMENT1`. Mereka dinamai demikian karena data sumber yang dimigrasi untuk dua file.csv ini dikaitkan dengan indeks segmen sumber berjalan yang sama. Di sisi lain, data migrasi yang disimpan di setiap target `SEGMENT2_LOAD00000009.csv` dan `SEGMENT3_LOAD0000000A.csv` file dikaitkan dengan indeks segmen sumber berjalan yang berbeda. Setiap file memiliki nama file yang diawali dengan nama indeks segmen yang sedang berjalan, `SEGMENT2` dan `SEGMENT3`.

Untuk jenis `ranges` beban paralel, Anda menentukan nama kolom dan nilai kolom menggunakan `columns` dan `boundaries` pengaturan aturan. `table-settings` Dengan aturan ini, Anda dapat menentukan partisi yang sesuai dengan nama segmen, sebagai berikut.

```
"parallel-load": {
  "type": "ranges",
```

```

    "columns": [
      "region",
      "sale"
    ],
    "boundaries": [
      [
        "NORTH",
        "1000"
      ],
      [
        "WEST",
        "3000"
      ]
    ],
    "segment-names": [
      "custom_segment1",
      "custom_segment2",
      "custom_segment3"
    ]
  }

```

Di sini, `segment-names` pengaturan mendefinisikan nama untuk tiga partisi untuk memigrasikan data secara paralel pada target S3. Data yang dimigrasi dimuat secara paralel dan disimpan dalam `file.csv` di bawah subfolder partisi secara berurutan, sebagai berikut.

```

.../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

```

Di sini, AWS DMS menyimpan serangkaian `file.csv` di masing-masing dari tiga subfolder partisi. Rangkaian `file.csv` di setiap subfolder partisi diberi nama secara bertahap mulai dari `LOAD00000001.csv` hingga semua data dimigrasikan.

Dalam beberapa kasus, Anda mungkin tidak secara eksplisit memberi nama subfolder partisi untuk jenis `ranges` beban paralel menggunakan pengaturan `segment-names`. Dalam hal ini, AWS DMS terapkan default untuk membuat setiap seri `file.csv` di bawah subfoldernya `table_name`. Di sini, AWS DMS awalan nama file dari setiap seri `file.csv` dengan nama indeks segmen sumber berjalan, sebagai berikut.

```

.../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
```

Pengaturan titik akhir saat menggunakan Amazon S3 sebagai target untuk AWS DMS

Anda dapat menggunakan pengaturan titik akhir untuk mengonfigurasi basis data target Amazon S3 mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir target menggunakan AWS DMS konsol, atau dengan menggunakan create-endpoint perintah di [AWS CLI](#), dengan sintaks `--s3-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan titik akhir yang dapat Anda gunakan dengan Amazon S3 sebagai target.

Opsi	Deskripsi
CsvNullValue	<p>Parameter opsional yang menentukan bagaimana AWS DMS memperlakukan nilai null. Saat menangani nilai null, Anda dapat menggunakan parameter ini untuk meneruskan string yang ditetapkan pengguna sebagai null saat menulis ke target. Misalnya, ketika kolom target dapat dibatalkan, Anda dapat menggunakan opsi ini untuk membedakan antara nilai string kosong dan nilai null. Jadi, jika Anda menetapkan nilai parameter ini ke string kosong ("" atau ""), AWS DMS memperlakukan string kosong sebagai nilai null, bukan. NULL</p> <p>Nilai default: NULL</p> <p>Nilai valid: string apapun yang benar</p> <p>Contoh: <code>--s3-settings '{"CsvNullValue": " "}'</code></p>
AddColumnName	<p>Parameter opsional yang ketika diatur ke <code>true</code> atau <code>y</code>, dapat Anda gunakan untuk menambahkan informasi nama kolom ke file keluaran <code>.csv</code>.</p> <p>Anda tidak dapat menggunakan parameter ini dengan <code>PreserveTransactions</code> atau <code>CdcPath</code>.</p> <p>Nilai default: <code>false</code></p>

Opsi	Deskripsi
	<p>Nilai valid: true, false, y, n</p> <p>Contoh: <code>--s3-settings '{"AddColumnName": true}'</code></p>
AddTrailingPaddingCharacter	<p>Gunakan pengaturan titik akhir target S3 AddTrailingPaddingCharacter untuk menambahkan padding pada data string. Nilai default-nya adalah false.</p> <p>Tipe: Boolean</p> <p>Contoh: <code>--s3-settings '{"AddTrailingPaddingCharacter": true}'</code></p>
BucketFolder	<p>Parameter opsional untuk menentukan nama folder dalam bucket S3. Jika tersedia, objek target dibuat sebagai file .csv atau .parquet di jalur <i>BucketFolder /schema_name /table_name /</i>. Jika parameter ini tidak ditentukan, maka jalur yang digunakan adalah <i>schema_name /table_name /</i>.</p> <p>Contoh: <code>--s3-settings '{"BucketFolder": "testFolder"}'</code></p>
BucketName	<p>Nama bucket S3 di mana objek target S3 dibuat sebagai file .csv atau .parquet.</p> <p>Contoh: <code>--s3-settings '{"BucketName": "buckettest"}'</code></p>

Opsi	Deskripsi
CannedAc1For0bjects	<p>Nilai yang memungkinkan AWS DMS untuk menentukan daftar kontrol akses yang telah ditetapkan sebelumnya (terekam) untuk objek yang dibuat di bucket S3 sebagai file .csv atau .parquet. Untuk informasi lebih lanjut tentang ACL terekam Amazon S3, lihat ACL Terekam di Panduan Developer Amazon S3.</p> <p>Nilai default: NONE</p> <p>Nilai valid untuk atribut ini adalah: NONE; PRIVATE; PUBLIC_READ_WRITE; AUTHENTICATED_READ; AWS_EXEC_READ; BUCKET_OWNER_READ; BUCKET_OWNER_FULL_CONTROL.</p> <p>Contoh: <code>--s3-settings '{"CannedAc1For0bjects": "PUBLIC_READ"}'</code></p>

Opsi	Deskripsi
CdcInsertsOnly	<p>Parameter opsional selama beban pengambilan data perubahan (CDC) beban hanya bisa menulis operasi INSERT ke file output comma-separated value (.csv) atau penyimpanan berbentuk kolom (.parquet). Secara default (Pengaturan <code>false</code>), field pertama dalam catatan .csv atau .parquet berisi huruf I (INSERT), U (UPDATE), atau D (DELETE). Huruf ini menunjukkan apakah baris dimasukkan, diperbarui, atau dihapus pada basis data sumber untuk beban CDC ke target. Jika <code>cdcInsertsOnly</code> diatur ke <code>true</code> atau <code>y</code>, hanya INSERT dari basis data sumber yang bermigrasi ke file .csv atau .parquet.</p> <p>Untuk format .csv saja, cara pencatatan INSERT ini bergantung pada nilai <code>IncludeOpForFullLoad</code>. Jika <code>IncludeOpForFullLoad</code> diatur ke <code>true</code>, field pertama dari setiap catatan CDC diatur ke I untuk menunjukkan operasi INSERT pada sumber. Jika <code>IncludeOpForFullLoad</code> diatur ke <code>false</code>, setiap catatan CDC ditulis tanpa field pertama untuk menunjukkan operasi INSERT pada sumber. Untuk informasi lebih lanjut tentang cara parameter ini bekerja sama, lihat Menunjukkan operasi sumber DB dalam data S3 bermigrasi.</p> <p>Nilai default: <code>false</code></p> <p>Nilai valid: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p> <p>Contoh: <code>--s3-settings '{"CdcInsertsOnly": true}'</code></p>

Opsi	Deskripsi
CdcInsertsAndUpdates	<p>Mengaktifkan beban perubahan pengambilan data (CDC) untuk menulis operasi INSERT dan UPDATE untuk file keluaran .csv atau .parquet (penyimpanan berbentuk kolom). Pengaturan default adalah false, tetapi ketika <code>cdcInsertsAndUpdates</code> diatur ke <code>true</code> atau <code>y</code>, INSERT dan UPDATE dari basis data sumber dimigrasi ke file .csv atau .parquet.</p> <p>Khusus format file .csv, cara pencatatan INSERT dan UPDATE ini tergantung pada nilai parameter <code>includeOpForFullLoad</code> . Jika <code>includeOpForFullLoad</code> diatur ke <code>true</code>, field pertama dari setiap catatan CDC diatur baik ke I maupun U untuk menunjukkan operasi INSERT dan UPDATE pada sumber. Tetapi jika <code>includeOpForFullLoad</code> diatur ke <code>false</code>, catatan CDC ditulis tanpa indikasi operasi INSERT atau UPDATE pada sumber.</p> <p>Untuk informasi lebih lanjut tentang cara parameter ini bekerja sama, lihat Menunjukkan operasi sumber DB dalam data S3 bermigrasi.</p> <div data-bbox="472 1037 1507 1400" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p><code>CdcInsertsOnly</code> dan <code>cdcInsertsAndUpdates</code> keduanya tidak dapat diatur ke <code>true</code> untuk titik akhir yang sama. Tetapkan salah satu antara <code>cdcInsertsOnly</code> atau <code>cdcInsertsAndUpdates</code> ke <code>true</code> untuk titik akhir yang sama, tapi tidak bisa keduanya.</p></div> <p>Nilai default: <code>false</code></p> <p>Nilai valid: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p> <p>Contoh: <code>--s3-settings '{"CdcInsertsAndUpdates": true}'</code></p>

Opsi	Deskripsi
CdcPath	<p>Menentukan jalur folder file CDC. Untuk sumber S3, pengaturan ini diperlukan jika sebuah tugas menangkap perubahan data; jika tidak, itu opsional. Jika CdcPath diatur, DMS membaca file CDC dari jalur ini dan mereplikasi perubahan data ke titik akhir target. Untuk target S3, jika Anda menetapkan <code>PreserveTransactions</code> ke <code>true</code>, DMS memverifikasi bahwa Anda telah menetapkan parameter ini ke jalur folder pada target S3 Anda tempat DMS dapat menyimpan urutan transaksi untuk beban CDC. DMS membuat jalur folder CDC ini di direktori kerja target S3 Anda atau lokasi target S3 ditentukan oleh <code>BucketFolder</code> dan <code>BucketName</code> .</p> <p>Anda tidak dapat menggunakan parameter ini dengan <code>DatePartitionEnabled</code> atau <code>AddColumnName</code> .</p> <p>Tipe: String</p> <p>Misalnya, jika Anda menentukan CdcPath sebagai <code>MyChangedData</code> , dan Anda menetapkan <code>BucketName</code> sebagai <code>MyTargetBucket</code> tapi tidak menentukan <code>BucketFolder</code> , DMS akan menciptakan jalur folder CDC berikut: <code>MyTargetBucket/MyChangedData</code> .</p> <p>Jika Anda menentukan CdcPath yang sama, dan Anda menetapkan <code>BucketName</code> sebagai <code>MyTargetBucket</code> dan <code>BucketFolder</code> sebagai <code>MyTargetData</code> , DMS membuat jalur folder CDC berikut: <code>MyTargetBucket/MyTargetData/MyChangedData</code> .</p> <div data-bbox="472 1388 1507 1843"><p> Note</p><p>Pengaturan ini didukung dalam AWS DMS versi 3.4.2 dan yang lebih tinggi.</p><p>Saat menangkap perubahan data dalam urutan transaksi, DMS selalu menyimpan perubahan baris dalam file.csv terlepas dari nilai pengaturan <code>DataFormat</code> S3 pada target. DMS tidak menyimpan perubahan data dalam urutan transaksi menggunakan file .parquet.</p></div>

Opsi	Deskripsi
CdcMaxBatchInterval	<p>Kondisi panjang interval maksimum, didefinisikan dalam detik, untuk keluaran file ke Amazon S3.</p> <p>Nilai default: 60 detik</p> <p>Kapan <code>CdcMaxBatchInterval</code> ditentukan dan <code>CdcMinFileSize</code> ditentukan, penulisan file dipicu oleh kondisi parameter mana pun yang terpenuhi terlebih dahulu.</p>
CdcMinFileSize	<p>Kondisi ukuran file minimum seperti yang didefinisikan dalam kilobyte untuk menampilkan file ke Amazon S3.</p> <p>Nilai default: 32000 KB</p> <p>Kapan <code>CdcMinFileSize</code> ditentukan dan <code>CdcMaxBatchInterval</code> ditentukan, penulisan file dipicu oleh kondisi parameter mana pun yang terpenuhi terlebih dahulu.</p>
PreserveTransactions	<p>Jika diatur ke <code>true</code>, DMS menyimpan urutan transaksi untuk perubahan pengambilan data (CDC) pada target Amazon S3 yang ditentukan oleh <code>CdcPath</code>.</p> <p>Anda tidak dapat menggunakan parameter ini dengan <code>DatePartitionEnabled</code> atau <code>AddColumnName</code>.</p> <p>Tipe: Boolean</p> <p>Saat menangkap perubahan data dalam urutan transaksi, DMS selalu menyimpan perubahan baris dalam file.csv terlepas dari nilai pengaturan <code>DataFormat S3</code> pada target. DMS tidak menyimpan perubahan data dalam urutan transaksi menggunakan file .parquet.</p> <div data-bbox="472 1598 1507 1814" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Pengaturan ini didukung dalam AWS DMS versi 3.4.2 dan yang lebih tinggi.</p> </div>

Opsi	Deskripsi
IncludeOpForFullLoad	<p>Parameter opsional selama beban penuh untuk menulis operasi INSERT hanya pada file keluaran comma-separated value (.csv).</p> <p>Untuk beban penuh, catatan hanya bisa dimasukkan. Secara default (pengaturan <code>false</code>), tidak ada informasi yang dicatat ke dalam file keluaran ini bagi beban penuh untuk menunjukkan bahwa baris dimasukkan pada basis data sumber. Jika <code>IncludeOpForFullLoad</code> diatur ke <code>true</code> atau <code>y</code>, INSERT dicatat sebagai penjelasan I di field pertama dari file .csv.</p> <div data-bbox="472 667 1508 1033" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>Parameter ini bekerja sama dengan <code>CdcInsertsOnly</code> atau <code>CdcInsertsAndUpdates</code> untuk keluaran ke file .csv saja. Untuk informasi lebih lanjut tentang cara parameter ini bekerja sama, lihat Menunjukkan operasi sumber DB dalam data S3 bermigrasi.</p> </div> <p>Nilai default: <code>false</code></p> <p>Nilai valid: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p> <p>Contoh: <code>--s3-settings '{"IncludeOpForFullLoad": true}'</code></p>
CompressionType	<p>Parameter opsional ketika diatur ke GZIP menggunakan GZIP untuk mengompresi file target .csv atau .parquet. Ketika parameter ini diatur ke default, file tidak terkompresi.</p> <p>Nilai default: NONE</p> <p>Nilai yang valid: GZIP atau NONE</p> <p>Contoh: <code>--s3-settings '{"CompressionType": "GZIP}"</code></p>

Opsi	Deskripsi
CsvDelimiter	<p>Pembatas yang digunakan untuk memisahkan kolom dalam file sumber .csv. Default adalah koma (,).</p> <p>Contoh: <code>--s3-settings '{"CsvDelimiter": ","}'</code></p>
CsvRowDelimiter	<p>Pembatas yang digunakan untuk memisahkan baris dalam file sumber .csv. Default adalah baris baru (\n).</p> <p>Contoh: <code>--s3-settings '{"CsvRowDelimiter": "\n"}'</code></p>
MaxFileSize	<p>Nilai yang menentukan ukuran maksimum (dalam KB) dari file.csv apapun yang akan dibuat saat bermigrasi ke target S3 selama pemuatan penuh.</p> <p>Nilai default: 1.048.576 KB (1 GB)</p> <p>Nilai valid: 1-1.048.576</p> <p>Contoh: <code>--s3-settings '{"MaxFileSize": 512}'</code></p>
Rfc4180	<p>Parameter opsional yang digunakan untuk mengatur perilaku agar sesuai dengan RFC untuk data yang dimigrasi ke Amazon S3 menggunakan format file .csv saja. Ketika nilai ini disetel ke <code>true</code> atau <code>y</code> menggunakan Amazon S3 sebagai target, jika data memiliki tanda kutip, koma, atau karakter baris baru di dalamnya, AWS DMS lampirkan seluruh kolom dengan pasangan tambahan tanda kutip ganda ("). Setiap tanda kutip dalam data diulang dua kali. Format ini sesuai dengan RFC 4180.</p> <p>Nilai default: <code>true</code></p> <p>Nilai valid: <code>true, false, y, n</code></p> <p>Contoh: <code>--s3-settings '{"Rfc4180": false}'</code></p>

Opsi	Deskripsi
EncryptionMode	<p>Mode enkripsi sisi server yang Anda inginkan untuk mengenkripsi file .csv atau .parquet objek disalin ke S3. Nilai valid adalah SSE_S3 (enkripsi sisi server S3) atau SSE_KMS (enkripsi kunci KMS). Jika Anda memilih SSE_KMS, atur parameter <code>ServerSideEncryptionKmsKeyId</code> Untuk Amazon Resource Name (ARN) untuk kunci KMS yang akan digunakan untuk enkripsi.</p> <div data-bbox="472 541 1507 909" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>Anda juga dapat menggunakan perintah CLI <code>modify-endpoint</code> untuk mengubah nilai atribut <code>EncryptionMode</code> untuk titik akhir yang sudah ada dari SSE_KMS ke SSE_S3. Tapi Anda tidak dapat mengubah nilai <code>EncryptionMode</code> dari SSE_S3 ke SSE_KMS.</p> </div> <p>Nilai default: SSE_S3</p> <p>Nilai yang valid: SSE_S3 atau SSE_KMS</p> <p>Contoh: <code>--s3-settings '{"EncryptionMode": SSE_S3}'</code></p>
ServerSideEncryptionKmsKeyId	<p>Jika Anda menyetel <code>EncryptionMode</code> ke SSE_KMS, setel parameter ini ke Amazon Resource Name (ARN) untuk kunci KMS. Anda dapat menemukan ARN ini dengan memilih alias kunci dalam daftar kunci AWS KMS yang dibuat untuk akun Anda. Saat membuat kunci, Anda harus mengaitkan kebijakan dan peran tertentu yang terkait dengan kunci KMS ini. Untuk informasi selengkapnya, lihat Membuat kunci AWS KMS untuk mengenkripsi objek target Amazon S3.</p> <p>Contoh: <code>--s3-settings '{"ServerSideEncryptionKmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/11a1a1a1-aaaa-9999-abab-2bbbbbb222a2"}'</code></p>

Opsi	Deskripsi
DataFormat	<p>Format keluaran untuk file yang digunakan oleh AWS DMS untuk membuat objek S3. Untuk target Amazon S3, AWS DMS mendukung baik file .csv maupun .parquet. File .parquet memiliki format penyimpanan kolom biner dengan pilihan kompresi yang efisien dan performa kueri yang lebih cepat. Untuk informasi lebih lanjut tentang file .parquet https://parquet.apache.org/.</p> <p>Nilai default: csv</p> <p>Nilai yang valid: csv atau parquet</p> <p>Contoh: <code>--s3-settings '{"DataFormat": "parquet"}</code></p>
EncodingType	<p>Jenis pengodean Parquet. Pilihan tipe pengodean mencakup hal berikut:</p> <ul style="list-style-type: none">• <code>rle-dictionary</code> – pengodean kamus ini menggunakan kombinasi pengodean bit-packing dan run-length untuk lebih efisien dalam menyimpan nilai yang berulang.• <code>plain</code> – Tidak ada pengodean.• <code>plain-dictionary</code> – Pengodean kamus ini membangun sebuah kamus nilai yang ditemui dalam kolom tertentu. Kamus disimpan dalam halaman kamus untuk setiap potongan kolom. <p>Nilai default: <code>rle-dictionary</code></p> <p>Nilai valid: <code>rle-dictionary</code> , <code>plain</code>, atau <code>plain-dictionary</code></p> <p>Contoh: <code>--s3-settings '{"EncodingType": "plain-dictionary"}</code></p>

Opsi	Deskripsi
DictPageSizeLimit	<p>Ukuran maksimum yang diizinkan, dalam byte, untuk halaman kamus dalam file .parquet. Jika halaman kamus melebihi nilai ini, maka halaman menggunakan pengodean biasa.</p> <p>Nilai default: 1.024.000 (1 MB)</p> <p>Nilai valid: Setiap nilai integer yang valid</p> <p>Contoh: <code>--s3-settings '{"DictPageSizeLimit": 2,048,000}'</code></p>
RowGroupLength	<p>Jumlah baris dalam satu grup baris dari file .parquet.</p> <p>Nilai default: 10.024 (10 KB)</p> <p>Nilai yang valid: Setiap bilangan bulat yang valid</p> <p>Contoh: <code>--s3-settings '{"RowGroupLength": 20,048}'</code></p>
DataPageSize	<p>Ukuran maksimum yang diizinkan, dalam byte, untuk halaman data dalam file .parquet.</p> <p>Nilai default: 1.024.000 (1 MB)</p> <p>Nilai yang valid: Setiap bilangan bulat yang valid</p> <p>Contoh: <code>--s3-settings '{"DataPageSize": 2,048,000}'</code></p>
ParquetVersion	<p>Versi format file .parquet.</p> <p>Nilai default: PARQUET_1_0</p> <p>Nilai yang valid: PARQUET_1_0 atau PARQUET_2_0</p> <p>Contoh: <code>--s3-settings '{"ParquetVersion": "PARQUET_2_0"}</code></p>

Opsi	Deskripsi
<p><code>EnableStatistics</code></p>	<p>Atur ke <code>true</code> atau <code>y</code> untuk mengaktifkan statistik tentang halaman file <code>.parquet</code> dan grup barisan.</p> <p>Nilai default: <code>true</code></p> <p>Nilai valid: <code>true, false, y, n</code></p> <p>Contoh: <code>--s3-settings '{"EnableStatistics": false}'</code></p>
<p><code>TimestampColumnName</code></p>	<p>Parameter opsional untuk menyertakan kolom stempel waktu dalam data titik akhir target S3.</p> <p>AWS DMS menyertakan STRING kolom tambahan dalam file objek <code>.csv</code> atau <code>.parquet</code> dari data yang dimigrasi saat Anda menyetel <code>TimestampColumnName</code> ke nilai yang tidak kosong.</p> <p>Untuk beban penuh, setiap baris dari kolom stempel waktu ini berisi stempel waktu untuk ketika data ditransfer dari sumber ke target oleh DMS.</p> <p>Untuk beban CDC, setiap baris kolom stempel waktu berisi stempel waktu untuk menjalankan baris tersebut dalam basis data sumber.</p> <p>Format string untuk nilai kolom stempel waktu ini adalah <code>yyyy-MM-dd HH:mm:ss.SSSSSS</code>. Secara default, presisi nilai ini dalam mikrodetik. Untuk beban CDC, pembulatan presisi tergantung pada stempel waktu melakukan didukung oleh DMS untuk basis data sumber.</p> <p>Ketika <code>AddColumnName</code> parameter diatur ke <code>true</code>, DMS juga menyertakan nama untuk kolom stempel waktu yang Anda tetapkan sebagai nilai tidak kosong. <code>TimestampColumnName</code></p> <p>Contoh: <code>--s3-settings '{"TimestampColumnName": "TIMESTAMP"}</code></p>

Opsi	Deskripsi
UseTaskStartTimeForFullLoadTimestamp	<p>Ketika diatur ke <code>true</code>, parameter ini menggunakan waktu mulai tugas sebagai nilai kolom stempel data alih-alih data waktu ditulis ke target. Untuk beban penuh, ketika <code>UseTaskStartTimeForFullLoadTimestamp</code> diatur ke <code>true</code>, setiap baris dari kolom stempel waktu berisi waktu mulai tugas. Untuk beban CDC, setiap baris kolom stempel waktu berisi waktu melakukan transaksi.</p> <p>Saat <code>UseTaskStartTimeForFullLoadTimestamp</code> diatur ke <code>false</code>, stempel waktu beban penuh di kolom stempel waktu bertambah dengan data waktu tiba di target.</p> <p>Nilai default: <code>false</code></p> <p>Nilai valid: <code>true</code>, <code>false</code></p> <p>Contoh: <code>--s3-settings '{"UseTaskStartTimeForFullLoadTimestamp": true}'</code></p> <p><code>UseTaskStartTimeForFullLoadTimestamp: true</code> membantu membuat target S3 <code>TimestampColumnName</code> untuk beban penuh dapat diurutkan dengan <code>TimestampColumnName</code> untuk beban CDC.</p>

Ops	Deskripsi
ParquetTimestampInMillisecond	<p>Parameter opsional yang menentukan ketepatan dari nilai kolom <code>TIMESTAMP</code> mana pun yang ditulis ke file objek S3 dalam format <code>.parquet</code>.</p> <p>Ketika atribut ini diatur ke <code>true</code> atau <code>y</code>, AWS DMS menulis semua kolom <code>TIMESTAMP</code> dalam file berformat <code>.parquet</code> dengan presisi mili detik. Jika tidak, DMS menuliskan mereka dengan presisi mikro detik.</p> <p>Saat ini, Amazon Athena dan hanya AWS Glue dapat menangani presisi milidetik untuk <code>TIMESTAMP</code> nilai. Atur atribut ini ke <code>true</code> untuk file objek titik akhir S3 berformat <code>.parquet</code> hanya jika Anda berencana untuk membuat kueri atau memproses data dengan Athena atau AWS Glue.</p> <div data-bbox="472 800 1507 1192" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><ul style="list-style-type: none">• AWS DMS menulis apapun nilai kolom <code>TIMESTAMP</code> yang ditulis ke file S3 dalam format <code>.csv</code> dengan presisi mikro detik.• Pengaturan atribut ini tidak berpengaruh pada format string dari nilai kolom stempel waktu yang dimasukkan dengan menetapkan atribut <code>TimestampColumnName</code> .</div> <p>Nilai default: <code>false</code></p> <p>Nilai valid: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p> <p>Contoh: <code>--s3-settings '{"ParquetTimestampInMillisecond": true}'</code></p>

Opsi	Deskripsi
GlueCatalogGeneration	<p>Untuk menghasilkan AWS Glue Data Catalog, atur pengaturan titik akhir ini ke <code>true</code>.</p> <p>Nilai default: <code>false</code></p> <p>Nilai yang valid: <code>true</code>, <code>false</code>,</p> <p>Contoh: <code>--s3-settings '{"GlueCatalogGeneration": true}'</code></p> <p>Catatan: Jangan gunakan <code>GlueCatalogGeneration</code> dengan <code>PreserveTransactions</code> dan <code>CdcPath</code>.</p>

Menggunakan AWS Glue Data Catalog dengan target Amazon S3 untuk AWS DMS

AWS Glue adalah layanan yang menyediakan cara sederhana untuk mengkategorikan data, dan terdiri dari repositori metadata yang dikenal sebagai AWS Glue Data Catalog. Anda dapat berintegrasi AWS Glue Data Catalog dengan titik akhir target Amazon S3 dan kueri data Amazon S3 melalui layanan lain AWS seperti Amazon Athena. Amazon Redshift berfungsi dengan AWS Glue tetapi AWS DMS tidak mendukungnya sebagai opsi pra-bangun.

Untuk menghasilkan katalog data, atur pengaturan `GlueCatalogGeneration` endpoint ke `true`, seperti yang ditunjukkan pada AWS CLI contoh berikut.

```
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}'
```

Untuk tugas replikasi beban penuh yang menyertakan data csv tipe, setel `IncludeOpForFullLoad` ke `true`.

Jangan gunakan `GlueCatalogGeneration` dengan `PreserveTransactions` dan `CdcPath`. `AWS GlueCrawler` tidak dapat merekonsiliasi skema berbeda dari file yang disimpan di bawah yang ditentukan. `CdcPath`

Agar Amazon Athena dapat mengindeks data Amazon S3 Anda, dan agar Anda dapat melakukan kueri data menggunakan kueri SQL standar melalui Amazon Athena, peran IAM yang dilampirkan ke titik akhir harus memiliki kebijakan berikut:

```
{
  "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_*"
}
]
```

References

- Untuk informasi selengkapnya tentang AWS Glue, lihat [Konsep](#) dalam Panduan Developer AWS Glue.
- Untuk informasi selengkapnya, AWS Glue Data Catalog lihat [Komponen](#) dalam Panduan AWS Glue Pengembang.

Menggunakan enkripsi data, file parquet, dan CDC pada target Amazon S3 Anda

Anda dapat menggunakan pengaturan target titik akhir S3 pengaturan untuk mengonfigurasi hal berikut:

- Kunci KMS khusus untuk mengenkripsi objek target S3 Anda.
- Parquet file sebagai format penyimpanan untuk objek target S3.
- Penangkapan perubahan data (CDC) termasuk transaksi order pada target S3.
- Integrasikan AWS Glue Data Catalog dengan titik akhir target Amazon S3 Anda dan kueri data Amazon S3 melalui layanan lain seperti Amazon Athena.

Pengaturan kunci AWS KMS untuk enkripsi data

Contoh berikut menunjukkan konfigurasi kunci KMS kustom untuk mengenkripsi objek target S3 Anda. Untuk memulai, Anda dapat menjalankan perintah `create-endpoint` CLI berikut.

```
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"}'
```

Di sini, objek JSON ditentukan oleh pilihan `--s3-settings` yang mendefinisikan dua parameter. Satu berupa parameter `EncryptionMode` dengan nilai `SSE_KMS`. Yang lainnya berupa parameter `ServerSideEncryptionKmsKeyId` dengan nilai `arn:aws:kms:us-east-1:111122223333:key/72abb6fb-1e49-4ac1-9aed-c803dfcc0480`. Nilai ini adalah Amazon Resource Name (ARN) untuk kunci KMS kustom Anda. Untuk target S3, Anda juga menentukan pengaturan tambahan. Ini mengidentifikasi peran akses server, menyediakan pembatas untuk format penyimpanan objek CSV default, dan memberikan lokasi dan nama bucket untuk menyimpan objek target S3.

Secara default, enkripsi data S3 terjadi menggunakan enkripsi pada sisi server S3. Untuk target S3 di contoh sebelumnya, ini juga setara dengan menentukan pengaturan titik akhir seperti dalam contoh berikut.

```
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"}'
```

Untuk informasi lebih lanjut tentang bekerja dengan enkripsi sisi server S3, lihat [Melindungi Data Menggunakan Enkripsi Sisi Server](#).

Note

Anda juga dapat menggunakan perintah CLI `modify-endpoint` untuk mengubah nilai parameter `EncryptionMode` untuk titik akhir yang sudah ada dari `SSE_KMS` ke `SSE_S3`. Tapi Anda tidak dapat mengubah nilai `EncryptionMode` dari `SSE_S3` ke `SSE_KMS`.

Pengaturan dalam menggunakan file `.parquet` untuk menyimpan objek target S3

Format default untuk membuat objek target S3 adalah file `.csv`. Contoh berikut menunjukkan beberapa pengaturan titik akhir untuk menentukan file `.parquet` sebagai format untuk membuat objek target S3. Anda dapat menentukan format file `.parquet` dengan semua default, seperti dalam contoh berikut.

```
aws dms create-endpoint --endpoint-identifier s3-target-endpoint --engine-name s3 --
endpoint-type target
--s3-settings '{"ServiceAccessRoleArn": "your-service-access-ARN", "DataFormat":
"parquet"}'
```

Di sini, parameter `DataFormat` diatur ke `parquet` untuk mengaktifkan format dengan semua default S3. Default ini juga termasuk pengodean kamus (`"EncodingType": "rle-dictionary"`) yang menggunakan kombinasi pengodean bit-packing dan run-length untuk lebih efisien menyimpan nilai-nilai berulang.

Anda dapat menambahkan pengaturan tambahan untuk pilihan selain default seperti dalam contoh berikut.

```
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 }'
```

Di sini, selain parameter untuk beberapa pilihan bucket S3 standar dan parameter `DataFormat`, parameter file `.parquet` tambahan berikut ditetapkan:

- `EncodingType` – Atur ke pengodean kamus (`plain-dictionary`) yang menyimpan nilai-nilai yang ditemui di setiap kolom dalam potongan perkolom dari halaman kamus.

- `DictPageSizeLimit` – Atur ke ukuran halaman kamus maksimum 3 MB.
- `EnableStatistics` – Nonaktifkan default yang memungkinkan pengumpulan statistik tentang halaman file Parquet dan grup baris.

Perubahan pengambilan data (CDC) termasuk transaksi order pada target S3

Secara default ketika AWS DMS menjalankan tugas CDC, ia menyimpan semua perubahan baris yang dicatat dalam basis data sumber Anda (atau dalam beberapa basis data) dalam satu atau lebih file untuk setiap tabel. Setiap kumpulan file yang berisi perubahan untuk tabel yang sama berada di direktori target tunggal yang terkait dengan tabel tersebut. AWS DMS menciptakan direktori target sebanyak tabel basis data yang dimigrasi ke titik akhir target Amazon S3. File disimpan pada target S3 di direktori ini tanpa memperhatikan pesanan transaksi. Untuk informasi lebih lanjut tentang ketentuan penamaan file, konten data, dan format, lihat [Menggunakan Amazon S3 sebagai target untuk AWS Database Migration Service](#).

Untuk menangkap perubahan basis data sumber dengan cara yang sekaligus menangkap urutan transaksi, Anda dapat menentukan pengaturan titik akhir S3 yang mengarahkan AWS DMS untuk menyimpan perubahan baris untuk Semua tabel basis data dalam satu atau beberapa file.csv yang sudah dibuat, tergantung pada ukuran transaksi. File transaksi .csv ini berisi semua perubahan baris terdaftar secara berurutan dalam urutan transaksi untuk semua tabel yang terlibat dalam setiap transaksi. File-file transaksi ini diletakkan bersama dalam satu Direktori transaksi yang Anda juga tentukan pada target S3. Dalam setiap file transaksi, operasi transaksi dan identitas database dan tabel sumber untuk setiap perubahan baris disimpan sebagai bagian dari data baris sebagai berikut.

```
operation,table_name,database_schema_name,field_value,...
```

Di sini, *operation* adalah operasi transaksi pada baris yang diubah, *table_name* adalah nama tabel basis data di mana baris diubah, *database_schema_name* adalah nama skema basis data tempat tabel berada, dan *field_value* adalah yang pertama dari satu atau lebih nilai field yang menentukan data untuk baris.

Contoh berikut dari file transaksi menunjukkan baris yang diubah untuk satu atau beberapa transaksi yang melibatkan dua tabel.

```
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
```

Di sini, operasi transaksi pada setiap baris ditunjukkan oleh I (insert), U(update), atau D (delete) di kolom pertama. Nama tabel adalah nilai kolom kedua (misalnya, Names_03cdcad11a). Nama skema basis data adalah nilai kolom ketiga (misalnya, rdsTempsdb). Dan kolom yang tersisa diisi dengan data baris Anda sendiri (misalnya, 13, Daniel).

Selain itu, AWS DMS menamai file transaksi yang dibuat pada target Amazon S3 menggunakan stempel waktu sesuai dengan konvensi penamaan berikut.

```
CDC_TXN-timestamp.csv
```

Di sini, *timestamp* adalah waktu ketika file transaksi dibuat, seperti dalam contoh berikut.

```
CDC_TXN-20201117153046033.csv
```

Stempel waktu dalam nama file ini memastikan bahwa file transaksi dibuat dan tercantum dalam urutan transaksi ketika Anda mencantumkannya di direktori transaksi mereka.

Note

Saat menangkap perubahan data dalam urutan transaksi, AWS DMS selalu menyimpan perubahan baris dalam file .csv terlepas dari nilai pengaturan DataFormat S3 pada target. AWS DMS tidak menyimpan perubahan data dalam pesanan transaksi menggunakan file .parquet.

Untuk mengontrol frekuensi penulisan ke target Amazon S3 selama tugas replikasi data, Anda dapat mengonfigurasi dan pengaturan. `CdcMaxBatchInterval` `CdcMinFileSize` Hal ini dapat menghasilkan performa yang lebih baik ketika menganalisis data tanpa operasi overhead tambahan. Untuk informasi lebih lanjut, lihat [Pengaturan titik akhir saat menggunakan Amazon S3 sebagai target untuk AWS DMS](#)

Untuk memberitahu AWS DMS agar menyimpan semua perubahan baris dalam urutan transaksi

1. Setel pengaturan `PreserveTransactions` S3 pada target ke `true`.
2. Setel pengaturan `CdcPath` S3 pada target ke jalur folder relatif tempat Anda ingin AWS DMS menyimpan file .csv transaksi.

AWS DMS membuat jalur ini baik di bawah bucket target S3 default dan direktori kerja atau di folder bucket dan bucket yang Anda tentukan menggunakan pengaturan `BucketName` dan `BucketFolder` S3 pada target.

Menunjukkan operasi sumber DB dalam data S3 bermigrasi

Saat AWS DMS memigrasikan catatan ke target S3, itu dapat membuat field tambahan di setiap catatan yang dimigrasikan. Field tambahan ini menunjukkan operasi diterapkan ke catatan pada basis data sumber. Cara AWS DMS membuat dan menetapkan bidang pertama ini tergantung pada jenis tugas migrasi dan pengaturan `includeOpForFullLoad`, `cdcInsertsOnly`, dan `cdcInsertsAndUpdates`.

Untuk beban penuh saat `includeOpForFullLoad` adalah `true`, AWS DMS selalu buat bidang pertama tambahan di setiap catatan.csv. Field ini berisi huruf I (INSERT) untuk menunjukkan bahwa baris dimasukkan pada basis data sumber. Untuk beban CDC ketika `cdcInsertsOnly` `false` (default), AWS DMS juga selalu membuat bidang pertama tambahan di setiap catatan.csv atau .parquet. Field ini berisi huruf I (INSERT), U (UPDATE), atau D (DELETE) untuk menunjukkan apakah baris dimasukkan, diperbarui, atau dihapus pada sumber basis data.

Dalam tabel berikut, Anda dapat melihat bagaimana pengaturan `cdcInsertsOnly` atribut `includeOpForFullLoad` dan bekerja sama untuk memengaruhi pengaturan catatan yang dimigrasi.

Dengan pengaturan parameter ini		DMS menetapkan catatan target sebagai berikut untuk keluaran.csv dan .parquet	
<code>includeOpForFullLoad</code>	<code>cdcInsertsOnly</code>	Untuk beban penuh	Untuk beban CDC
<code>true</code>	<code>true</code>	Menambahkan nilai field pertama yang ditetapkan ke I	Menambahkan nilai field pertama yang ditetapkan ke I
<code>false</code>	<code>false</code>	Tidak ada field tambahan	Menambahkan nilai field pertama yang ditetapkan ke I, U, atau D

Dengan pengaturan parameter ini		DMS menetapkan catatan target sebagai berikut untuk keluaran.csv dan .parquet	
<code>includeOpForFullLoad</code>	<code>cdcInsertsOnly</code>	Untuk beban penuh	Untuk beban CDC
<code>false</code>	<code>true</code>	Tidak ada field tambahan	Tidak ada field tambahan
<code>true</code>	<code>false</code>	Menambahkan nilai field pertama yang ditetapkan ke I	Menambahkan nilai field pertama yang ditetapkan ke I, U, atau D

Saat `includeOpForFullLoad` dan `cdcInsertsOnly` ditetapkan ke nilai yang sama, catatan target diatur menurut atribut yang mengontrol pengaturan catatan untuk jenis migrasi saat ini. Atribut tersebut adalah `includeOpForFullLoad` untuk beban penuh dan `cdcInsertsOnly` untuk beban CDC.

Saat `includeOpForFullLoad` dan `cdcInsertsOnly` ditetapkan ke nilai yang berbeda, AWS DMS membuat pengaturan catatan target konsisten untuk CDC dan beban penuh. Hal ini dilakukan dengan membuat pengaturan catatan untuk beban CDC sesuai dengan pengaturan catatan untuk semua beban penuh sebelumnya yang ditentukan oleh `includeOpForFullLoad`.

Dengan kata lain, anggaplah beban penuh diatur untuk menambahkan field pertama untuk menunjukkan catatan yang dimasukkan. Dalam kasus ini, beban CDC berikut diatur untuk menambahkan field pertama yang menunjukkan catatan yang dimasukkan, diperbarui, atau dihapus sesuai sumbernya. Sebaliknya, anggaplah beban penuh diatur untuk tidak menambahkan field pertama untuk menunjukkan catatan yang dimasukkan. Dalam kasus ini, beban CDC juga diatur untuk tidak menambahkan field pertama untuk setiap catatan, terlepas dari operasi catatan yang sesuai sumbernya.

Demikian juga dengan bagaimana DMS membuat dan menetapkan field pertama tambahan bergantung pada pengaturan `includeOpForFullLoad` dan `cdcInsertsAndUpdates`. Dalam tabel berikut, Anda dapat melihat bagaimana pengaturan atribut `includeOpForFullLoad` dan `cdcInsertsAndUpdates` bekerja sama untuk memengaruhi pengaturan catatan yang dimigrasikan dalam format ini.

Dengan pengaturan parameter ini		DMS menetapkan catatan target sebagai berikut untuk keluaran .csv	
includeOpForFullLoad	cdcInsertsAndUpdate	Untuk beban penuh	Untuk beban CDC
true	true	Menambahkan nilai field pertama yang ditetapkan ke I	Menambahkan nilai field pertama yang ditetapkan ke I atau U
false	false	Tidak ada field tambahan	Menambahkan nilai field pertama yang ditetapkan ke I, U, atau D
false	true	Tidak ada field tambahan	Menambahkan nilai field pertama yang ditetapkan ke I atau U
true	false	Menambahkan nilai field pertama yang ditetapkan ke I	Menambahkan nilai field pertama yang ditetapkan ke I, U, atau D

Tipe data target untuk S3 Parquet

Tabel berikut menampilkan tipe data target Parquet yang didukung saat menggunakan AWS DMS dan pemetaan default dari tipe data AWS DMS.

Untuk informasi tambahan tentang tipe data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Tipe data AWS DMS	Tipe data parquet S3
BYTES	BINARY
DATE	DATE32
TIME	TIME32

Tipe data AWS DMS	Tipe data parquet S3
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
BOOLEAN	BOOL

Menggunakan basis data Amazon DynamoDB sebagai target untuk AWS Database Migration Service

Anda dapat menggunakan AWS DMS untuk memigrasikan data ke tabel Amazon DynamoDB. Amazon DynamoDB adalah layanan basis data NoSQL yang terkelola penuh yang menyediakan performa cepat dan dapat diprediksi dengan skalabilitas mulus. AWS DMS mendukung penggunaan basis data relasional atau MongoDB sebagai sumber.

Dalam DynamoDB, tabel, item, dan atribut adalah komponen inti yang Anda kerjakan. Sebuah tabel adalah kumpulan item, dan setiap item adalah kumpulan atribut. DynamoDB menggunakan kunci primer, yang disebut kunci partisi, untuk mengidentifikasi secara unik setiap item dalam sebuah tabel. Anda juga dapat menggunakan kunci dan indeks sekunder untuk memberikan lebih banyak fleksibilitas kueri.

Anda menggunakan pemetaan objek untuk memigrasi data Anda dari basis data sumber ke tabel DynamoDB target. Pemetaan objek mengizinkan Anda menentukan letak sumber data di target.

Saat AWS DMS membuat tabel pada titik akhir target DynamoDB, tabel akan dibuat sebanyak di titik akhir basis data sumber. AWS DMS juga menetapkan beberapa nilai parameter DynamoDB. Biaya pembuatan tabel tergantung pada jumlah data dan jumlah tabel yang akan dimigrasi.

Note

Opsi Mode SSL di AWS DMS konsol atau API tidak berlaku untuk beberapa streaming data dan layanan NoSQL seperti Kinesis dan DynamoDB. Mereka aman secara default, jadi AWS DMS menunjukkan pengaturan mode SSL sama dengan tidak ada (Mode SSL = Tidak Ada). Anda tidak perlu menyediakan konfigurasi tambahan untuk titik akhir Anda untuk menggunakan SSL. Misalnya, saat menggunakan DynamoDB sebagai titik akhir target, itu aman secara default. Semua panggilan API ke DynamoDB menggunakan SSL, jadi tidak perlu opsi SSL tambahan di titik akhir. AWS DMS Anda dapat menempatkan data dengan aman dan mengambil data melalui titik akhir SSL menggunakan protokol HTTPS, yang AWS DMS digunakan secara default saat menghubungkan ke database DynamoDB.

Untuk membantu meningkatkan kecepatan transfer, AWS DMS mendukung beban penuh multithread ke instans target DynamoDB. DMS mendukung multithreading ini dengan pengaturan tugas yang meliputi berikut ini:

- `MaxFullLoadSubTasks` - Gunakan pilihan ini untuk menunjukkan jumlah maksimum tabel sumber yang dimuat secara paralel. DMS memuat setiap tabel ke dalam tabel target DynamoDB yang sesuai menggunakan subtugas khusus. Nilai default adalah 8. Nilai maksimumnya adalah 49.
- `ParallelLoadThreads` - Gunakan pilihan ini untuk menentukan jumlah thread yang digunakan AWS DMS untuk memuat setiap tabel ke dalam tabel target DynamoDB-nya. Nilai default adalah 0 (single-threaded). Nilai maksimumnya adalah 200. Anda dapat meminta untuk meningkatkan batas maksimum ini.

 Note

DMS menetapkan setiap segmen tabel ke thread-nya sendiri untuk memuat. Oleh karena itu, atur `ParallelLoadThreads` untuk jumlah maksimum segmen yang Anda tentukan untuk tabel di sumber.

- `ParallelLoadBufferSize` - Gunakan pilihan ini untuk menentukan jumlah maksimum catatan yang disimpan dalam buffer yang menggunakan thread beban paralel untuk memuat data ke target DynamoDB. Nilai default adalah 50. Nilai maksimumnya adalah 1.000. Gunakan pengaturan ini dengan `ParallelLoadThreads`. `ParallelLoadBufferSize` hanya berlaku bila ada lebih dari satu thread.
- Pengaturan pemetaan tabel untuk tabel individual – Gunakan aturan `table-settings` untuk mengidentifikasi tabel individual dari sumber yang ingin Anda muat secara paralel. Juga gunakan aturan-aturan ini untuk menentukan bagaimana segmen baris setiap tabel dalam pemuatan multithread. Untuk informasi lebih lanjut, lihat [Tabel dan koleksi pengaturan aturan dan operasi](#).

 Note

Saat AWS DMS menetapkan nilai parameter DynamoDB untuk tugas migrasi, nilai parameter Read Capacity Unit (RCU) default diatur ke 200.

Nilai parameter Write Capacity Unit (WCU) juga ditetapkan, tetapi nilainya bergantung pada beberapa hal lain:

- Nilai default untuk parameter WCU adalah 200.
- Jika pengaturan tugas `ParallelLoadThreads` diatur lebih besar dari 1 (default adalah 0), maka parameter WCU diatur ke 200 dikali nilai `ParallelLoadThreads`.
- Standard AWS DMS biaya penggunaan berlaku untuk sumber daya yang Anda gunakan.

Migrasi dari basis data relasional ke tabel DynamoDB

AWS DMS mendukung memigrasikan data ke tipe data skalar DynamoDB. Saat bermigrasi dari basis data relasional seperti Oracle atau MySQL ke DynamoDB, Anda mungkin ingin merestrukturisasi cara Anda menyimpan data ini.

Saat ini AWS DMS mendukung restrukturisasi antar tabel tunggal ke atribut tipe skalar DynamoDB. Jika Anda memigrasikan data ke DynamoDB dari tabel basis data relasional, Anda mengambil data dari tabel dan melakukan format ulang menjadi atribut tipe data skalar DynamoDB. Atribut ini dapat menerima data dari beberapa kolom, dan Anda dapat memetakan kolom ke atribut secara langsung.

AWS DMS mendukung tipe data skalar DynamoDB berikut:

- String
- Number
- Boolean

Note

data NULL dari sumber diabaikan pada target.

Prasyarat untuk menggunakan DynamoDB sebagai target untuk AWS Database Migration Service

Sebelum Anda mulai bekerja dengan basis data DynamoDB sebagai target untuk AWS DMS, pastikan Anda membuat IAM role. Peran IAM ini harus memungkinkan AWS DMS untuk mengasumsikan dan memberikan akses ke tabel DynamoDB yang sedang dimigrasikan ke. Kumpulan izin akses minimum ditampilkan dalam kebijakan IAM berikut.

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

```

    },
    "Action": "sts:AssumeRole"
  }
]
}

```

Peran yang Anda gunakan untuk migrasi ke DynamoDB harus memiliki izin berikut.

```

{
  "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": "*"
    }
  ]
}

```

Keterbatasan saat menggunakan DynamoDB sebagai target untuk AWS Database Migration Service

Batasan berikut berlaku saat menggunakan DynamoDB sebagai target:

- DynamoDB membatasi ketepatan tipe data Number untuk 38 tempat. Menyimpan semua jenis data dengan presisi yang lebih tinggi sebagai String. Anda perlu secara eksplisit menentukan ini menggunakan fitur pemetaan objek.
- Karena DynamoDB tidak memiliki tipe data Date, data yang menggunakan tipe data Date dikonversi ke string.
- DynamoDB tidak mengizinkan pembaruan untuk atribut kunci primer. Pembatasan ini penting ketika menggunakan replikasi berkelanjutan dengan perubahan pengambilan data (CDC) karena dapat mengakibatkan data yang tidak diinginkan dalam target. Tergantung pada bagaimana Anda memiliki pemetaan objek, operasi CDC yang memperbarui kunci primer dapat melakukan salah satu dari dua hal. Yaitu bisa gagal atau memasukkan item baru dengan kunci primer diperbarui dan data yang tidak lengkap.
- AWS DMS hanya mendukung replikasi tabel dengan kunci primer nonkomposit. Pengecualiannya adalah jika Anda menentukan pemetaan objek untuk tabel target dengan tombol partisi kustom atau urutkan kunci, atau keduanya.
- AWS DMS tidak mendukung data LOB kecuali CLOB. AWS DMS mengkonversi data CLOB menjadi string DynamoDB saat migrasi data.
- Ketika Anda menggunakan DynamoDB sebagai target, hanya tabel kontrol Apply Exceptions (`dmslogs.aws_dms_apply_exceptions`) yang didukung. Untuk informasi lebih lanjut tentang tabel kontrol, lihat [Mengontrol pengaturan tugas tabel](#).
- AWS DMS tidak mendukung pengaturan tugas `TargetTablePrepMode=TRUNCATE_BEFORE_LOAD` untuk DynamoDB sebagai target.
- AWS DMS tidak mendukung pengaturan tugas `TaskRecoveryTableEnabled` untuk DynamoDB sebagai target.

Menggunakan pemetaan objek untuk migrasi data ke DynamoDB

AWS DMS menggunakan aturan pemetaan tabel untuk memetakan data dari sumber ke tabel target DynamoDB. Untuk memetakan data ke target DynamoDB, Anda menggunakan jenis aturan pemetaan tabel yang disebut Pemetaan objek. Pemetaan objek memungkinkan Anda menentukan nama atribut dan data yang akan dimigrasi ke mereka. Anda harus memiliki aturan seleksi ketika menggunakan pemetaan objek.

DynamoDB tidak memiliki struktur preset selain memiliki kunci partisi dan kunci pengurutan opsional. Jika Anda memiliki kunci primer nonkomposit, AWS DMS menggunakannya. Jika Anda memiliki kunci

primer komposit atau Anda ingin menggunakan urutan kunci, definisikan kunci-kunci tersebut dan atribut lainnya dalam tabel target DynamoDB Anda.

Untuk membuat aturan pemetaan objek, Anda menentukan `rule-type` sebagai Pemetaan objek. Aturan ini menentukan jenis pemetaan objek yang ingin Anda gunakan.

Struktur aturannya adalah sebagai berikut:

```
{ "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 saat ini mendukung `map-record-to-record` dan `map-record-to-document` sebagai satu-satunya nilai valid untuk parameter `rule-action`. Nilai-nilai ini menentukan apa yang AWS DMS lakukan secara default pada catatan yang tidak dikecualikan sebagai bagian dari daftar atribut `exclude-columns`. Nilai-nilai ini tidak mempengaruhi pemetaan atribut dengan cara apapun.

- Anda dapat menggunakan `map-record-to-record` saat migrasi dari basis data relasional ke DynamoDB. Menggunakan kunci primer dari basis data relasional sebagai kunci partisi di DynamoDB dan menciptakan atribut untuk setiap kolom dalam basis data sumber. Saat menggunakan `map-record-to-record`, untuk kolom mana pun dalam tabel sumber yang tidak tercantum dalam daftar atribut `exclude-columns`, AWS DMS menciptakan atribut yang sesuai pada instans target DynamoDB. Ia melakukannya terlepas dari apakah kolom sumber yang digunakan dalam pemetaan atribut.
- Anda menggunakan `map-record-to-document` untuk menempatkan kolom sumber ke dalam satu peta DynamoDB datar pada target menggunakan nama atribut “_doc.” Saat menggunakan `map-record-to-document`, AWS DMS menempatkan data ke dalam atribut satu peta DynamoDB datar pada sumber. Atribut ini disebut “_doc”. Penempatan ini berlaku untuk kolom mana pun di tabel sumber yang tidak tercantum dalam daftar atribut `exclude-columns`.

Salah satu cara untuk memahami perbedaan antara `rule-action` `map-record-to-record` dan `map-record-to-document` adalah untuk melihat dua parameter yang beraksi. Untuk contoh ini, anggap bahwa Anda memulai dengan baris tabel basis data relasional dengan struktur dan data berikut:

FirstName	LastName	NickName	WorkAddress	WorkPhone	HomeAddress	HomePhone	income
▶ Daniel	Sheridan	Dan	101 Main St Cambridge, MA	800-867-5309	100 Secret St, Unknowrville, MA	123-456-7890	12345678

Untuk memigrasikan informasi ini ke DynamoDB, Anda membuat aturan untuk memetakan data ke item tabel DynamoDB. Perhatikan kolom yang tercantum untuk parameter `exclude-columns`. Kolom ini tidak langsung dipetakan ke target. Sebagai gantinya, pemetaan atribut digunakan untuk menggabungkan data menjadi item baru, seperti di mana `FirstName` dan `LastName` dikelompokkan bersama untuk menjadi `CustomerName` target DynamoDB. `NickName` dan pendapatan tidak dikecualikan.

```
{
  "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",

```

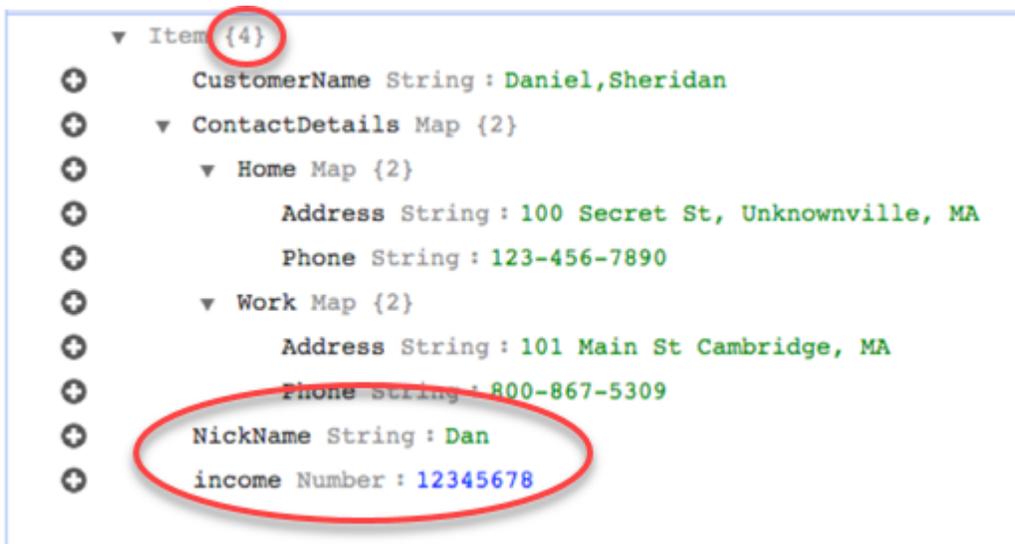
```

        "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}"
                            }
                        }
                    }
                }
            }
        }
    ]
}

```

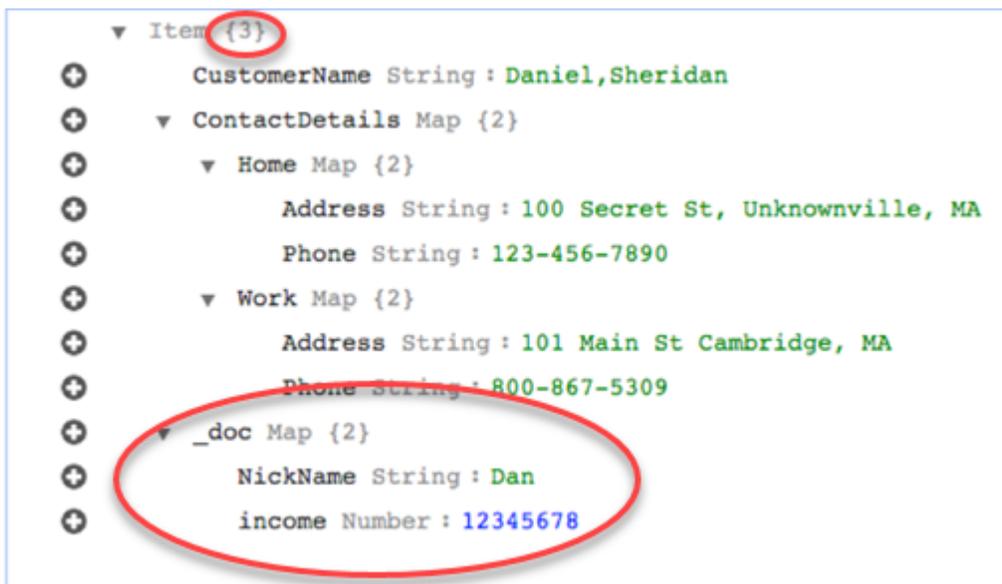
```
]
}
```

Dengan menggunakan rule-action parameter map-record-to-record, data untuk NickNamedan pendapatan dipetakan ke item dengan nama yang sama di target DynamoDB.



The screenshot shows a hierarchical view of a DynamoDB item. The root is 'Item {4}'. Underneath, there are several fields: 'CustomerName String : Daniel, Sheridan', 'ContactDetails Map {2}', 'Home Map {2}', 'Address String : 100 Secret St, Unknownville, MA', 'Phone String : 123-456-7890', 'Work Map {2}', 'Address String : 101 Main St Cambridge, MA', 'Phone String : 800-867-5309', 'NickName String : Dan', and 'income Number : 12345678'. The 'NickName String : Dan' and 'income Number : 12345678' fields are circled in red.

Namun, anggaplah Anda menggunakan aturan yang sama tetapi ubah rule-action parameternya menjadi map-record-to-document. Dalam hal ini, kolom yang tidak tercantum dalam exclude-columns parameter, NickNamedan pendapatan, dipetakan ke item _doc.



The screenshot shows a hierarchical view of a DynamoDB item. The root is 'Item {3}'. Underneath, there are several fields: 'CustomerName String : Daniel, Sheridan', 'ContactDetails Map {2}', 'Home Map {2}', 'Address String : 100 Secret St, Unknownville, MA', 'Phone String : 123-456-7890', 'Work Map {2}', 'Address String : 101 Main St Cambridge, MA', 'Phone String : 800-867-5309', and '_doc Map {2}'. The '_doc Map {2}' field is expanded to show 'NickName String : Dan' and 'income Number : 12345678', which are circled in red.

Menggunakan ekspresi kondisi kustom dengan pemetaan objek

Anda dapat menggunakan fitur DynamoDB yang disebut kondisi ekspresi untuk memanipulasi data yang sedang ditulis ke tabel DynamoDB. Untuk informasi lebih lanjut tentang kondisi ekspresi di DynamoDB, lihat [Kondisi ekspresi](#).

Sebuah anggota kondisi ekspresi terdiri dari:

- sebuah ekspresi (diperlukan)
- nilai atribut ekspresi (opsional). Menentukan struktur json DynamoDB dari nilai atribut
- nama atribut ekspresi (opsional)
- pilihan untuk kapan menggunakan kondisi ekspresi (opsional). Defaultnya adalah `apply-during-cdc = false` dan `apply-during-full-load = true`

Struktur aturannya adalah sebagai berikut:

```
"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>
    }
  }
```

Sampel berikut menyoroti bagian yang digunakan untuk kondisi ekspresi.



Menggunakan pemetaan atribut dengan pemetaan objek

Pemetaan atribut memungkinkan Anda menentukan string templat menggunakan nama kolom sumber untuk merestrukturisasi data pada target. Tidak ada pemformatan yang dilakukan selain yang ditentukan pengguna dalam templat.

Contoh berikut menunjukkan struktur basis data sumber dan struktur yang diinginkan dari target DynamoDB. Pertama ditunjukkan struktur sumbernya, dalam hal ini basis data Oracle, dan kemudian struktur data yang diinginkan di DynamoDB. Contoh diakhiri dengan JSON yang digunakan untuk membuat struktur target yang diinginkan.

Struktur data Oracle adalah sebagai berikut:

First	Last	Street	Home/Address	Home/Phone	Work/Address	Work/Phone	DateOfBirth
Kunci Primer				N/A			
Randy	Mar	5	221B Baker Street	1234567890	31 Spooner Street, Quahog	9876541230	02/29/1988

Struktur data DynamoDB adalah sebagai berikut:

Customer	StoreId	ContactDetails	DateOfBirth
Kunci partisi	Urutkan Kunci	N/A	
Randysh	5	<pre>{ "Name": "Randy", "Home": { "Address": "221B Baker Street", "Phone": 1234567890 }, "Work": { "Address": "31 Spooner Street, Quahog", "Phone": 9876541230 } }</pre>	02/29/1988

JSON berikut menunjukkan pemetaan objek dan pemetaan kolom yang digunakan untuk mencapai struktur 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}\"}}}"
      }
    ]
  }
}
]
}

```

Cara lain menggunakan pemetaan kolom adalah dengan menggunakan format DynamoDB sebagai jenis dokumen Anda. Contoh kode berikut menggunakan peta dynamodb sebagai attribute-sub-type untuk pemetaan atribut.

```

{
  "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}"
                }
            }
        }
    ]
}

```

Sebagai alternatif `dynamodb-map`, Anda dapat menggunakan `dynamodb-list` sebagai `attribute-sub-type` for pemetaan atribut, seperti yang ditunjukkan pada contoh berikut.

```

{
  "target-attribute-name": "ContactDetailsList",
  "attribute-type": "document",
  "attribute-sub-type": "dynamodb-list",
  "value": {
    "L": [
      {
        "N": "${FirstName}"
      },
      {
        "N": "${HomeAddress}"
      },
      {
        "N": "${HomePhone}"
      },
      {
        "N": "${WorkAddress}"
      }
    ]
  }
}

```

```

        "N": "${WorkPhone}"
    }
}
]
}
}

```

Contoh 1: Menggunakan pemetaan atribut dengan pemetaan objek

Contoh berikut memigrasikan data dari dua tabel database MySQL, `nfl_data` dan `sport_team`, ke dua tabel DynamoDB yang disebut `NFLTeams` dan `SportTeams`. Struktur tabel dan JSON yang digunakan untuk memetakan data dari tabel basis data MySQL ke tabel DynamoDB ditunjukkan sebagai berikut.

Struktur tabel basis data MySQL `nfl_data` ditunjukkan di bawah ini:

```
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    |       |
+-----+-----+-----+-----+-----+-----+
```

Struktur tabel basis data MySQL `sport_team` ditunjukkan di bawah ini:

```
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	

Aturan pemetaan tabel yang digunakan untuk memetakan dua tabel ke dua tabel DynamoDB adalah sebagai berikut:

```
{
  "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}\"}"
        }
      ]
    }
  }
}
]
}

```

Sampel keluaran untuk tabel DynamoDB NFITeams ditunjukkan di bawah ini:

```

  "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"
}
```

Output sampel untuk tabel SportsTeams DynamoDB ditunjukkan di bawah ini:

```
{
  "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"
}
```

Target tipe data untuk DynamoDB

Titik akhir DynamoDB untuk AWS DMS mendukung sebagian besar tipe data DynamoDB. Tabel berikut menunjukkan tipe data target Amazon AWS DMS yang didukung saat menggunakan AWS DMS dan pemetaan default dari tipe data AWS DMS.

Untuk informasi tambahan tentang tipe data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Saat AWS DMS memigrasi data dari basis data heterogen, kita memetakan tipe data dari basis data sumber ke tipe data perantara yang disebut tipe data AWS DMS. Kami kemudian memetakan tipe data perantara untuk tipe data target. Tabel berikut menunjukkan setiap tipe data AWS DMS dan tipe data yang dipetakannya di DynamoDB:

Tipe data AWS DMS	Tipe data DynamoDB
String	String
WString	String
Boolean	Boolean

Tipe data AWS DMS	Tipe data DynamoDB
Date	String
DateTime	String
INT1	Number
INT2	Number
INT4	Number
INT8	Number
Numeric	Number
Real4	Number
Real8	Number
UINT1	Number
UINT2	Number
UINT4	Number
UINT8	Number
CLOB	String

Menggunakan Amazon Kinesis Data Streams sebagai target AWS Database Migration Service

Anda dapat menggunakannya AWS DMS untuk memigrasikan data ke aliran data Amazon Kinesis. Amazon Kinesis data streams adalah bagian dari layanan Amazon Kinesis Data Streams. Anda dapat menggunakan Kinesis data stream untuk mengumpulkan dan memproses sejumlah besar catatan data secara langsung.

Kinesis data stream adalah serangkaian serpihan. Serpihan secara unik diidentifikasi urutan catatan data dalam stream. Untuk informasi lebih lanjut tentang serpihan di Amazon Kinesis Data Streams, lihat [Serpihan](#) di Panduan Developer Amazon Kinesis Data Streams.

AWS Database Migration Service menerbitkan catatan ke aliran data Kinesis menggunakan JSON. Selama konversi, AWS DMS membuat serial setiap catatan dari basis data sumber ke dalam pasangan atribut-nilai dalam format JSON atau format pesan JSON_UNFORMATTED. Format pesan JSON_UNFORMATTED adalah string JSON baris tunggal dengan pembatas baris baru. Ini memungkinkan Amazon Data Firehose untuk mengirimkan data Kinesis ke tujuan Amazon S3, dan kemudian menanyakannya menggunakan berbagai mesin kueri termasuk Amazon Athena.

Anda menggunakan pemetaan objek untuk memigrasi data Anda dari sumber data mana pun yang didukung ke aliran target. Dengan pemetaan objek, Anda menentukan bagaimana struktur catatan data dalam aliran. Anda juga menentukan kunci partisi untuk setiap tabel, yang digunakan Kinesis Data Streams untuk mengelompokkan data ke dalam serpihannya.

Saat AWS DMS membuat tabel pada titik akhir target Kinesis Data Streams, itu membuat tabel sebanyak di titik akhir database sumber. AWS DMS juga menetapkan beberapa nilai parameter Kinesis Data Streams. Biaya pembuatan tabel tergantung pada jumlah data dan jumlah tabel yang akan dimigrasi.

Note

Opsi Mode SSL di AWS DMS konsol atau API tidak berlaku untuk beberapa streaming data dan layanan NoSQL seperti Kinesis dan DynamoDB. Mereka aman secara default, jadi AWS DMS menunjukkan pengaturan mode SSL sama dengan tidak ada (Mode SSL = Tidak Ada). Anda tidak perlu menyediakan konfigurasi tambahan untuk titik akhir Anda untuk menggunakan SSL. Misalnya, saat menggunakan Kinesis sebagai titik akhir target, Kinesis sudah aman secara default. Semua panggilan API ke Kinesis menggunakan SSL, jadi tidak perlu opsi SSL tambahan di titik akhir. AWS DMS Anda dapat menempatkan data dengan aman dan mengambil data melalui titik akhir SSL menggunakan protokol HTTPS, yang AWS DMS digunakan secara default saat menghubungkan ke Kinesis Data Stream.

Pengaturan titik akhir Kinesis Data Streams

Saat Anda menggunakan titik akhir target Kinesis Data Streams, Anda bisa mendapatkan detail transaksi dan kontrol `KinesisSettings` menggunakan opsi di API. AWS DMS

Anda dapat mengatur pengaturan koneksi dengan cara berikut:

- Di AWS DMS konsol, menggunakan pengaturan titik akhir.
- Di CLI, menggunakan `kinesis-settings` opsi perintah. [CreateEndpoint](#)

Dalam CLI, gunakan parameter permintaan dari pilihan `kinesis-settings` berikut:

Note

Support untuk pengaturan titik akhir `IncludeNullAndEmpty` tersedia pada AWS DMS versi 3.4.1 dan setelahnya. Tetapi dukungan untuk pengaturan titik akhir berikut lainnya untuk target Kinesis Data Streams tersedia di AWS DMS

- `MessageFormat` - Format keluaran untuk catatan yang dibuat pada titik akhir. Format pesan adalah JSON (default) atau `JSON_UNFORMATTED` (satu baris tanpa tab).
- `IncludeControlDetails` - Menampilkan informasi kontrol terperinci untuk definisi tabel, definisi kolom, serta perubahan tabel dan kolom pada pesan keluaran Kinesis. Default adalah `false`.
- `IncludeNullAndEmpty` - Sertakan NULL dan kolom kosong dalam target. Default adalah `false`.
- `IncludePartitionValue` - Menampilkan nilai partisi dalam pesan keluaran Kinesis, kecuali tipe partisi `schema-table-type`. Default adalah `false`.
- `IncludeTableAlterOperations` - Termasuk setiap operasi bahasa definisi data (DDL) yang mengubah tabel dalam data kontrol, seperti `rename-table`, `drop-table`, `add-column`, `drop-column`, dan `rename-column`. Default adalah `false`.
- `IncludeTransactionDetails` - Menyediakan informasi transaksi detail dari basis data sumber. Informasi ini mencakup stempel waktu melakukan, posisi log, dan nilai-nilai untuk `transaction_id`, `previous_transaction_id`, dan `transaction_record_id` (catatan offset dalam transaksi). Default adalah `false`.
- `PartitionIncludeSchemaTable` - Skema prefiks dan nama tabel untuk nilai partisi, ketika tipe partisi `primary-key-type`. Melakukan hal ini meningkatkan distribusi data di antara serpihan Kinesis. Sebagai contoh, anggaplah bahwa skema `SysBench` memiliki ribuan tabel dan setiap tabel hanya memiliki rentang terbatas untuk kunci primer. Dalam kasus ini, kunci primer yang sama dikirim dari ribuan tabel ke serpihan yang sama, yang menyebabkan throttling. Default adalah `false`.

Contoh berikut menunjukkan bahwa kinesis-settings digunakan dengan sebuah contoh perintah `create-endpoint` yang dikeluarkan menggunakan perintah 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
```

Pengaturan tugas beban penuh multithread

Untuk membantu meningkatkan kecepatan transfer, AWS DMS mendukung beban penuh multithreaded ke instance target Kinesis Data Streams. DMS mendukung multithreading ini dengan pengaturan tugas yang meliputi berikut ini:

- `MaxFullLoadSubTasks` - Gunakan pilihan ini untuk menunjukkan jumlah maksimum tabel sumber yang dimuat secara paralel. DMS memuat setiap tabel ke dalam tabel target Kinesis yang sesuai menggunakan subtugas khusus. Default adalah 8; nilai maksimum adalah 49.
- `ParallelLoadThreads`— Gunakan opsi ini untuk menentukan jumlah utas yang AWS DMS digunakan untuk memuat setiap tabel ke dalam tabel target Kinesis. Nilai maksimum untuk target Kinesis Data Streams adalah 32. Anda dapat meminta untuk meningkatkan batas maksimum ini.
- `ParallelLoadBufferSize` - Gunakan pilihan ini untuk menentukan jumlah catatan maksimum yang disimpan di buffer yang digunakan oleh thread beban paralel untuk memuat data ke target Kinesis. Nilai default adalah 50. Nilai maksimumnya adalah 1.000. Gunakan pengaturan ini dengan `ParallelLoadThreads`. `ParallelLoadBufferSize` hanya berlaku bila ada lebih dari satu thread.
- `ParallelLoadQueuesPerThread` - Gunakan pilihan ini untuk menentukan jumlah antrean yang diakses setiap thread secara bersamaan untuk mengambil catatan data dari antrean dan menghasilkan beban batch untuk target. Default adalah 1. Tetapi untuk target Kinesis dengan berbagai ukuran muatan, rentang yang valid adalah 5-512 antrean per thread.

Pengaturan beban tugas CDC multithread

Anda dapat meningkatkan performa perubahan pengambilan data (CDC) untuk titik akhir target streaming data secara langsung seperti Kinesis menggunakan pengaturan tugas untuk mengubah

perilaku `PutRecords` Panggilan API. Untuk melakukan ini, Anda dapat menentukan jumlah thread yang terjadi bersamaan, antrean per thread, dan jumlah catatan yang disimpan dalam buffer menggunakan pengaturan tugas `ParallelApply*`. Misalnya, Anda ingin melakukan beban CDC dan menerapkan 128 thread secara paralel. Anda juga ingin mengakses 64 antrean per thread, dengan 50 catatan disimpan per buffer.

Untuk mempromosikan kinerja CDC, AWS DMS mendukung pengaturan tugas ini:

- `ParallelApplyThreads`— Menentukan jumlah thread bersamaan yang AWS DMS digunakan selama beban CDC untuk mendorong catatan data ke titik akhir target Kinesis. Nilai default adalah nol (0) dan nilai maksimum adalah 32.
- `ParallelApplyBufferSize` - Menentukan jumlah maksimum catatan yang disimpan di setiap antrean buffer untuk thread serentak untuk mendorong ke titik akhir target Kinesis selama beban CDC. Nilai default adalah 100 dan nilai maksimum adalah 1.000. Gunakan pilihan ini saat `ParallelApplyThreads` menentukan lebih dari satu thread.
- `ParallelApplyQueuesPerThread` - Menentukan jumlah antrean yang diakses oleh setiap thread untuk mengambil catatan data dari antrean dan menghasilkan beban batch untuk titik akhir Kinesis selama CDC. Nilai default adalah 1 dan nilai maksimum adalah 512.

Saat menggunakan pengaturan tugas `ParallelApply*`, default `partition-key-type`-nya adalah `primary-key` dari tabel, bukan `schema-name.table-name`.

Menggunakan citra sebelumnya untuk melihat nilai asli dari baris CDC untuk Kinesis data stream sebagai target

Ketika menulis pembaruan CDC ke target data-streaming seperti Kinesis Anda dapat melihat nilai asli baris basis data sumber sebelum diubah oleh pembaruan. Untuk memungkinkan hal ini, AWS DMS mengisi gambar sebelum peristiwa pembaruan berdasarkan data yang disediakan oleh mesin database sumber.

Mesin basis data sumber yang berbeda memberikan jumlah informasi yang berbeda untuk citra sebelum:

- Oracle menyediakan update untuk kolom hanya jika mereka berubah.
- PostgreSQL menyediakan data hanya untuk kolom yang merupakan bagian dari kunci primer (berubah atau tidak). Untuk menyediakan data untuk semua kolom (diubah atau tidak), Anda perlu mengatur `REPLICA_IDENTITY` ke `FULL` bukan `DEFAULT`. Perhatikan bahwa Anda harus memilih `REPLICA_IDENTITY` pengaturan dengan hati-hati untuk setiap tabel. Jika Anda mengatur

REPLICA_IDENTITY keFULL, semua nilai kolom ditulis ke write-ahead logging (WAL) terus menerus. Hal ini dapat menyebabkan masalah kinerja atau sumber daya dengan tabel yang sering diperbarui.

- MySQL umumnya menyediakan data untuk semua kolom kecuali untuk tipe data BLOB dan CLOB (diubah atau tidak).

Untuk mengaktifkan pencitraan sebelumnya dengan tujuan menambahkan nilai asli dari basis data sumber untuk keluaran AWS DMS, gunakan pengaturan tugas `BeforeImageSettings` atau parameter `add-before-image-columns`. Parameter ini menerapkan aturan transformasi kolom.

`BeforeImageSettings` menambahkan atribut JSON baru untuk setiap operasi pembaruan dengan nilai yang dikumpulkan dari sistem basis data sumber, seperti yang ditunjukkan berikut.

```
"BeforeImageSettings": {
  "EnableBeforeImage": boolean,
  "FieldName": string,
  "ColumnFilter": pk-only (default) / non-lob / all (but only one)
}
```

Note

Hanya berlaku `BeforeImageSettings` untuk AWS DMS tugas yang berisi komponen CDC, seperti beban penuh ditambah tugas CDC (yang memigrasi data yang ada dan mereplikasi perubahan yang sedang berlangsung), atau ke tugas hanya CDC (yang hanya mereplikasi perubahan data). Jangan terapkan `BeforeImageSettings` pada tugas-tugas yang beban penuh saja.

Untuk pilihan `BeforeImageSettings`, hal berikut berlaku:

- Atur pilihan `EnableBeforeImage` ke `true` untuk mengaktifkan pencitraan sebelum. Default adalah `false`.
- Gunakan pilihan `FieldName` untuk memberikan nama ke atribut JSON baru. Ketika `EnableBeforeImage` adalah `true`, `FieldName` diperlukan dan tidak dapat kosong.
- Pilihan `ColumnFilter` menentukan kolom untuk menambahkan dengan menggunakan pencitraan sebelumnya. Untuk menambahkan kolom yang hanya merupakan bagian dari kunci primer tabel

saja, gunakan nilai default, `pk-only`. Untuk menambahkan kolom apa pun yang memiliki nilai citra sebelum, gunakan `all`. Perhatikan bahwa gambar sebelum tidak berisi kolom dengan tipe data LOB, seperti CLOB atau BLOB.

```
"BeforeImageSettings": {
  "EnableBeforeImage": true,
  "FieldName": "before-image",
  "ColumnFilter": "pk-only"
}
```

Note

Target Amazon S3 tidak mendukung `BeforeImageSettings`. Untuk target S3, gunakan hanya aturan transformasi `add-before-image-columns` untuk melakukan pencitraan sebelumnya selama CDC.

Menggunakan aturan transformasi citra sebelumnya

Sebagai alternatif untuk pengaturan tugas, Anda dapat menggunakan parameter `add-before-image-columns`, yang menerapkan aturan transformasi kolom. Dengan parameter ini, Anda dapat mengaktifkan pencitraan sebelum selama CDC pada target data streaming seperti Kinesis.

Dengan menggunakan `add-before-image-columns` dalam aturan transformasi, Anda dapat menerapkan kontrol yang lebih halus dari hasil citra sebelum. Aturan transformasi memungkinkan Anda untuk menggunakan pencari objek yang memberikan Anda kontrol atas tabel yang dipilih untuk aturan tersebut. Dan juga, Anda dapat merangkai aturan transformasi, yang memungkinkan aturan yang berbeda untuk diterapkan ke tabel yang berbeda. Anda kemudian dapat memanipulasi kolom yang dihasilkan dengan menggunakan aturan lain.

Note

Jangan gunakan parameter `add-before-image-columns` bersamaan dengan pengaturan tugas `BeforeImageSettings` dalam tugas yang sama. Sebaliknya, gunakan salah satu saja, parameter atau pengaturan, tetapi tidak keduanya, untuk satu tugas.

Sebuah tipe peraturan transformation dengan parameter add-before-image-columns untuk kolom harus menyediakan before-image-def bagian. Berikut ini adalah sebuah contoh.

```
{
  "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,
  }
}
```

Nilai dari column-prefix ditambahkan sebelum nama kolom, dan nilai default column-prefix adalah BI_. Nilai dari column-suffix ditambahkan sebelum nama kolom, dan default kosong. Jangan atur column-prefix dan column-suffix ke string kosong.

Pilih satu nilai untuk column-filter. Untuk menambahkan kolom yang merupakan bagian dari kunci primer tabel saja, pilih pk-only. Untuk menambahkan kolom yang bukan tipe LOB saja, gunakan non-lob. Atau pilih all untuk menambahkan kolom yang memiliki nilai citra-sebelum.

Contoh untuk aturan transformasi citra sebelumnya

Aturan transformasi dalam contoh berikut menambahkan kolom baru yang disebut BI_emp_no pada target. Jadi pernyataan seperti UPDATE employees SET emp_no = 3 WHERE emp_no = 1; mengisi BI_emp_no field dengan 1. Ketika Anda menulis pembaruan CDC untuk target Amazon S3, kolom BI_emp_no memungkinkan untuk memberitahu baris asli yang diperbarui.

```
{
  "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"
  }
}
]
}

```

Untuk informasi tentang penggunaan peraturan tindakan `add-before-image-columns`, lihat [Aturan dan tindakan transformasi](#).

Prasyarat untuk menggunakan aliran data Kinesis sebagai target AWS Database Migration Service

Peran IAM untuk menggunakan aliran data Kinesis sebagai target AWS Database Migration Service

Sebelum Anda mengatur aliran data Kinesis sebagai target AWS DMS, pastikan Anda membuat peran IAM. Peran ini harus memungkinkan AWS DMS untuk mengasumsikan dan memberikan akses ke aliran data Kinesis yang sedang dimigrasikan ke. Kumpulan izin akses minimum ditampilkan dalam kebijakan IAM berikut.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "1",
      "Effect": "Allow",
      "Principal": {
        "Service": "dms.amazonaws.com"
      },
    },
  ],
}

```

```
"Action": "sts:AssumeRole"
}
]
}
```

Peran yang Anda gunakan untuk migrasi ke Kinesis data stream harus memiliki izin berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesis:DescribeStream",
        "kinesis:PutRecord",
        "kinesis:PutRecords"
      ],
      "Resource": "arn:aws:kinesis:region:accountID:stream/streamName"
    }
  ]
}
```

Mengakses aliran data Kinesis sebagai target AWS Database Migration Service

Di AWS DMS versi 3.4.7 dan yang lebih tinggi, untuk terhubung ke titik akhir Kinesis, Anda harus melakukan salah satu hal berikut:

- Konfigurasi DMS untuk menggunakan titik akhir VPC. Untuk informasi tentang mengonfigurasi DMS untuk menggunakan titik akhir VPC, lihat. [Mengkonfigurasi titik akhir VPC sebagai AWSSumber DMS dan titik akhir target](#)
- Konfigurasi DMS untuk menggunakan rute publik, yaitu membuat instance replikasi Anda menjadi publik. Untuk informasi tentang contoh replikasi publik, lihat. [Instans replikasi publik dan privat](#)

Keterbatasan saat menggunakan Kinesis Data Streams sebagai target AWS Database Migration Service

Batasan berikut berlaku saat menggunakan Kinesis Data Streams sebagai target:

- AWS DMS menerbitkan setiap pembaruan ke satu catatan dalam database sumber sebagai satu catatan data dalam aliran data Kinesis tertentu terlepas dari transaksi. Namun, Anda dapat menyertakan detail transaksi untuk setiap data catatan dengan menggunakan parameter yang relevan dari API `KinesisSettings`.
- Mode LOB penuh tidak didukung.
- Ukuran LOB maksimum yang didukung adalah 1 MB.
- Kinesis Data Streams tidak mendukung deduplikasi. Aplikasi yang mengonsumsi data dari aliran perlu menangani duplikasi catatan. Untuk informasi lebih lanjut, lihat [Menangani catatan duplikat](#) di Panduan Developer Amazon Kinesis Data Streams.
- AWS DMS mendukung dua bentuk berikut untuk kunci partisi:
 - `SchemaName.TableName`: Kombinasi skema dan nama tabel.
 - `${AttributeName}`: Nilai salah satu field di JSON, atau kunci primer dari tabel dalam basis data sumber.
- Untuk informasi tentang mengenkripsi data at rest Anda saat berada dalam Kinesis Data Streams, lihat [Perlindungan data dalam Kinesis Data Streams](#) di Panduan Developer AWS Key Management Service .
- `BatchApply` tidak didukung untuk titik akhir Kinesis. Menggunakan penerapan Batch (misalnya, pengaturan tugas metadata target `BatchApplyEnabled`) untuk target Kinesis dapat mengakibatkan hilangnya data.
- Target kinesis hanya didukung untuk aliran data Kinesis di AWS akun yang sama dan sama dengan instance replikasi Wilayah AWS .
- Saat bermigrasi dari sumber MySQL, `BeforeImage` data tidak menyertakan tipe data CLOB dan BLOB. Untuk informasi selengkapnya, lihat [Menggunakan citra sebelumnya untuk melihat nilai asli dari baris CDC untuk Kinesis data stream sebagai target](#).
- AWS DMS tidak mendukung migrasi nilai tipe `BigInt` data dengan lebih dari 16 digit. Untuk mengatasi batasan ini, Anda dapat menggunakan aturan transformasi berikut untuk mengonversi `BigInt` kolom menjadi string. Untuk informasi selengkapnya tentang aturan transformasi, lihat [Aturan dan tindakan transformasi](#).

```
{
```

```

"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
}
}

```

Menggunakan pemetaan objek untuk memigrasi data ke Kinesis data stream

AWS DMS menggunakan aturan pemetaan tabel untuk memetakan data dari sumber ke aliran data Kinesis target. Untuk memetakan data ke aliran target, Anda menggunakan jenis aturan pemetaan tabel yang disebut pemetaan objek. Anda menggunakan pemetaan objek untuk menentukan cara pencatatan data di peta sumber ke pencatatan data yang dipublikasikan ke Kinesis data stream.

Kinesis data stream tidak memiliki struktur preset selain memiliki kunci partisi. Dalam aturan pemetaan objek, nilai-nilai yang mungkin dari `partition-key-type` untuk catatan data adalah `schema-table`, `transaction-id`, `primary-key`, `constant`, dan `attribute-name`.

Untuk membuat aturan pemetaan objek, Anda tentukan `rule-type` sebagai `object-mapping`. Aturan ini menentukan jenis pemetaan objek yang ingin Anda gunakan.

Struktur untuk aturan tersebut adalah sebagai berikut.

```

{
  "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 saat ini mendukung `map-record-to-record` dan `map-record-to-document` sebagai satu-satunya nilai yang valid untuk `rule-action` parameter. Pengaturan ini memengaruhi nilai yang tidak dikecualikan sebagai bagian dari daftar `exclude-columns` atribut. `map-record-to-document` Nilai `map-record-to-record` dan menentukan bagaimana AWS DMS menangani catatan ini secara default. Nilai-nilai ini tidak mempengaruhi pemetaan atribut dengan cara apapun.

Gunakan `map-record-to-record` saat bermigrasi dari basis data relasional ke Kinesis data stream. Jenis aturan ini menggunakan nilai `taskResourceId.schemaName.tableName` dari basis data relasional sebagai kunci partisi dalam Kinesis data stream dan membuat atribut untuk setiap kolom dalam basis data sumber.

Saat menggunakan `map-record-to-record`, perhatikan hal berikut:

- Pengaturan ini hanya memengaruhi kolom yang dikecualikan oleh `exclude-columns` daftar.
- Untuk setiap kolom tersebut, AWS DMS buat atribut yang sesuai dalam topik target.
- AWS DMS menciptakan atribut yang sesuai ini terlepas dari apakah kolom sumber digunakan dalam pemetaan atribut.

Gunakan `map-record-to-document` untuk menempatkan kolom sumber ke dalam satu, dokumen datar dalam aliran target yang sesuai menggunakan nama atribut `"_doc"`. AWS DMS menempatkan data ke dalam satu peta datar pada sumber yang disebut `"_doc"`. Penempatan ini berlaku untuk kolom mana pun di tabel sumber yang tidak tercantum dalam daftar atribut `exclude-columns`.

Salah satu cara untuk memahami `map-record-to-record` adalah melihatnya beraksi. Untuk contoh ini, anggaplah bahwa Anda memulai dengan baris tabel basis data relasional dengan struktur dan data berikut.

FirstName	LastName	StoreId	HomeAddress	HomePhone	WorkAddress	WorkPhone	DateofBirth
Randy	Marsh	5	221B Baker Street	1234567890	31 Spooner	9876543210	02/29/1988

FirstName	LastName	StoreId	HomeAddress	HomePhone	WorkAddress	WorkPhone	DateofBirth
					Street, Quahog		

Untuk memigrasi informasi ini dari skema bernama Test ke Kinesis data stream, Anda membuat aturan untuk memetakan data ke aliran target. Aturan berikut menggambarkan pemetaan.

```
{
  "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"
      }
    }
  ]
}
```

Berikut ini menggambarkan format catatan yang dihasilkan dalam Kinesis data stream:

- StreamName: XXX
- PartitionKey: Test.Customers //schmaname.TableName
- Data: //Pesan JSON berikut

```
{
  "FirstName": "Randy",
  "LastName": "Marsh",
  "StoreId": "5",
  "HomeAddress": "221B Baker Street",
  "HomePhone": "1234567890",
  "WorkAddress": "31 Spooner Street, Quahog",
  "WorkPhone": "9876543210",
  "DateOfBirth": "02/29/1988"
}
```

Namun, anggaplah Anda menggunakan aturan yang sama tetapi mengubah parameter `rule-action` untuk `map-record-to-document` dan mengecualikan kolom tertentu. Aturan berikut menggambarkan pemetaan.

```
{
  "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"
    ]
  }
}
]
}

```

Dalam kasus ini, kolom tidak tercantum dalam parameter `exclude-columns`, `FirstName`, `LastName`, `StoreId` dan `DateOfBirth`, dipetakan ke `_doc`. Berikut ini menggambarkan format catatan yang dihasilkan.

```

{
  "data":{
    "_doc":{
      "FirstName": "Randy",
      "LastName": "Marsh",
      "StoreId": "5",
      "DateOfBirth": "02/29/1988"
    }
  }
}

```

Restrukturisasi data dengan pemetaan atribut

Anda dapat merestrukturisasi data saat memigrasinya ke Kinesis data stream menggunakan peta atribut. Misalnya, Anda mungkin ingin menggabungkan beberapa field di sumber menjadi field tunggal dalam target. Peta atribut berikut menggambarkan bagaimana merestrukturisasi data.

```

{
  "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}"
                    }
                }
            }
        ]
    }
}

```


Format pesan untuk Kinesis Data Streams

Keluaran JSON hanya berupa daftar pasangan nilai-kunci. Format pesan JSON_UNFORMATTED adalah string JSON baris tunggal dengan pembatas baris baru.

AWS DMS menyediakan bidang cadangan berikut untuk mempermudah penggunaan data dari Kinesis Data Streams:

RecordType

Jenis catatan dapat berupa data atau kontrol. Catatan data mewakili baris yang sebenarnya dalam sumber. Kontrol catatan adalah untuk peristiwa-peristiwa penting di aliran, misalnya restart tugas.

Operasi

Untuk catatan data, operasinya dapat load, insert, update, atau delete.

Untuk catatan kontrol, operasi dapat berupa create-table, rename-table, drop-table, change-columns, add-column, drop-column, rename-column, atau column-type-change.

SchemaName

Skema sumber untuk catatan. Field ini bisa kosong untuk catatan kontrol.

TableName

Tabel sumber untuk catatan. Field ini bisa kosong untuk catatan kontrol.

Stempel waktu

Stempel waktu untuk saat pesan JSON dibuat. Field ini diformat dengan format ISO 8601.

Menggunakan Apache Kafka sebagai target AWS Database Migration Service

Anda dapat menggunakan AWS DMS untuk memigrasikan data ke cluster Apache Kafka. Apache Kafka adalah platform streaming terdistribusi. Anda dapat menggunakan Apache Kafka untuk menelan dan memproses data streaming secara langsung.

AWS juga menawarkan Amazon Managed Streaming for Apache Kafka (Amazon MSK) untuk digunakan sebagai target. AWS DMS Amazon MSK adalah layanan streaming Apache Kafka yang

terkelola penuh yang menyederhanakan implementasi dan pengelolaan instans Apache Kafka. Ini bekerja dengan versi Apache Kafka open-source, dan Anda mengakses instans MSK Amazon sebagai AWS DMS target persis seperti instans Apache Kafka lainnya. Untuk informasi lebih lanjut, lihat [Apa itu Amazon MSK?](#) dalam Panduan Developer: Amazon Managed Streaming for Apache Kafka.

Sebuah kluster Kafka menyimpan aliran catatan dalam kategori yang disebut topik yang dibagi menjadi partisi. Partisi adalah urutan catatan data (pesan) yang diidentifikasi secara unik secara unik dalam suatu topik. Partisi dapat didistribusikan ke beberapa broker dalam sebuah kluster untuk mengaktifkan pemrosesan paralel catatan topik. Untuk informasi lebih lanjut tentang topik dan partisi dan distribusi mereka di Apache Kafka, lihat [Topik dan log](#) dan [Distribusi](#).

Kluster Kafka Anda dapat berupa instans MSK Amazon, kluster yang berjalan di instans Amazon EC2, atau kluster lokal. Instans MSK Amazon atau cluster pada instans Amazon EC2 dapat berada di VPC yang sama atau yang berbeda. Jika kluster Anda lokal, Anda dapat menggunakan server nama lokal Anda sendiri untuk instance replikasi untuk menyelesaikan nama host kluster. Untuk informasi tentang menyiapkan server nama untuk instance replikasi Anda, lihat [Menggunakan server nama on-premise Anda sendiri](#). Untuk informasi selengkapnya tentang pengaturan jaringan, lihat [Menyiapkan jaringan untuk instans replikasi](#).

Saat menggunakan kluster MSK Amazon, pastikan grup keamanannya mengizinkan akses dari instance replikasi Anda. Untuk informasi tentang mengubah grup keamanan untuk kluster MSK Amazon, lihat [Mengubah grup keamanan kluster MSK Amazon](#).

AWS Database Migration Service menerbitkan catatan ke topik Kafka menggunakan JSON. Selama konversi, AWS DMS membuat serial setiap catatan dari basis data sumber ke dalam pasangan atribut-nilai dalam format JSON.

Untuk memigrasikan data Anda dari sumber data apa pun yang didukung ke kluster Kafka target, Anda menggunakan pemetaan objek. Dengan pemetaan objek, Anda menentukan bagaimana struktur catatan data dalam topik target. Anda juga menentukan kunci partisi untuk setiap tabel, yang digunakan Apache Kafka untuk mengelompokkan data ke dalam partisi.

Saat ini, AWS DMS mendukung satu topik per tugas. Untuk satu tugas dengan beberapa tabel, semua pesan masuk ke satu topik. Setiap pesan menyertakan bagian metadata yang mengidentifikasi skema dan tabel target. AWS DMS versi 3.4.6 dan yang lebih tinggi mendukung replikasi multitopik menggunakan pemetaan objek. Untuk informasi selengkapnya, lihat [Replikasi multitopik menggunakan pemetaan objek](#).

Pengaturan titik akhir Apache Kafka

Anda dapat menentukan detail koneksi melalui pengaturan titik akhir di AWS DMS konsol, atau `--kafka-settings` opsi di CLI. Persyaratan untuk setiap pengaturan:

- **Broker** - Tentukan lokasi dari satu atau lebih broker di klaster Kafka Anda dalam bentuk daftar yang dipisahkan koma untuk masing-masing *broker-hostname:port*. Contohnya `"ec2-12-345-678-901.compute-1.amazonaws.com:2345,ec2-10-987-654-321.compute-1.amazonaws.com:2345"`. Pengaturan ini dapat menentukan lokasi dari salah satu atau semua broker dalam klaster. Semua Klaster broker berkomunikasi untuk menangani partisi catatan data yang dimigrasikan ke topik.
- **Topic** - (Opsional) Tentukan nama topik dengan panjang maksimum 255 huruf dan simbol. Anda dapat menggunakan titik (`.`), garis bawah (`_`), dan minus (`-`). Nama topik dengan titik (`.`) atau garis bawah (`_`) dapat bertabrakan dalam struktur data internal. Gunakan salah satu, tetapi jangan kedua simbol ini sekaligus dalam nama topik. Jika Anda tidak menentukan nama topik, AWS DMS gunakan `"kafka-default-topic"` sebagai topik migrasi.

Note

Untuk AWS DMS membuat topik migrasi yang Anda tentukan atau topik default, tetapkan `auto.create.topics.enable = true` sebagai bagian dari konfigurasi cluster Kafka Anda. Untuk informasi lebih lanjut, lihat [Keterbatasan saat menggunakan Apache Kafka sebagai target AWS Database Migration Service](#)

- **MessageFormat** - Format keluaran untuk catatan yang dibuat pada titik akhir. Format pesan adalah JSON (default) atau JSON_UNFORMATTED (satu baris tanpa tab).
- **MessageMaxBytes** - Ukuran maksimum dalam byte untuk catatan yang dibuat pada titik akhir. Default adalah 1.000.000.

Note

Anda hanya dapat menggunakan AWS CLI/SDK untuk mengubah `MessageMaxBytes` ke nilai non-default. Misalnya, untuk memodifikasi titik akhir Kafka yang sudah ada dan mengubah `MessageMaxBytes`, gunakan perintah berikut.

```
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** - Menyediakan informasi transaksi detail dari basis data sumber. Informasi ini mencakup stempel waktu melakukan, posisi log, dan nilai-nilai untuk `transaction_id`, `previous_transaction_id`, dan `transaction_record_id` (catatan offset dalam transaksi). Default adalah `false`.
- **IncludePartitionValue** - Menampilkan nilai partisi dalam pesan keluaran Kafka, kecuali jika tipe partisi adalah `schema-table-type`. Default adalah `false`.
- **PartitionIncludeSchemaTable** - Skema prefiks dan nama tabel untuk nilai partisi, ketika tipe partisi `primary-key-type`. Melakukan hal ini meningkatkan distribusi data antara partisi Kafka. Sebagai contoh, anggaplah bahwa skema SysBench memiliki ribuan tabel dan setiap tabel hanya memiliki rentang terbatas untuk kunci primer. Dalam kasus ini, kunci primer yang sama dikirim dari ribuan tabel ke partisi yang sama, sehingga menyebabkan throttling. Default adalah `false`.
- **IncludeTableAlterOperations** - Termasuk setiap operasi bahasa definisi data (DDL) yang mengubah tabel dalam data kontrol, seperti `rename-table`, `drop-table`, `add-column`, `drop-column`, dan `rename-column`. Default adalah `false`.
- **IncludeControlDetails** - Menunjukkan informasi kontrol detail untuk definisi tabel, definisi kolom, dan perubahan tabel dan kolom pada keluaran pesan Kafka. Default adalah `false`.
- **IncludeNullAndEmpty** - Sertakan NULL dan kolom kosong dalam target. Default adalah `false`.
- **SecurityProtocol** - Menyetel koneksi yang aman ke titik akhir target Kafka menggunakan Keamanan Lapisan Pengangkutan (TLS). Pilihan termasuk `ssl-authentication`, `ssl-encryption`, dan `sasl-ssl`. Menggunakan `sasl-ssl` membutuhkan `SaslUsername` dan `SaslPassword`.
- **SslEndpointIdentificationAlgorithm**— Menetapkan verifikasi nama host untuk sertifikat. Pengaturan ini didukung di AWS DMS versi 3.5.1 dan yang lebih baru. Opsi mencakup hal berikut:
 - **NONE**: Nonaktifkan verifikasi nama host broker dalam koneksi klien.
 - **HTTPS**: Aktifkan verifikasi nama host broker dalam koneksi klien.

Anda dapat menggunakan pengaturan untuk membantu meningkatkan kecepatan transfer Anda. Untuk melakukannya, AWS DMS mendukung beban penuh multithread klaster target Apache Kafka. AWS DMS mendukung multithreading dengan pengaturan tugas yang meliputi hal berikut ini:

- `MaxFullLoadSubTasks`— Gunakan opsi ini untuk menunjukkan jumlah maksimum tabel sumber untuk dimuat secara paralel. AWS DMS memuat setiap tabel ke dalam tabel target Kafka yang sesuai menggunakan subtugas khusus. Default adalah 8; nilai maksimum adalah 49.
- `ParallelLoadThreads`— Gunakan opsi ini untuk menentukan jumlah utas yang AWS DMS digunakan untuk memuat setiap tabel ke dalam tabel target Kafka. Nilai maksimum untuk target Apache Kafka adalah 32. Anda dapat meminta untuk meningkatkan batas maksimum ini.
- `ParallelLoadBufferSize` - Gunakan pilihan ini untuk menentukan jumlah maksimum catatan untuk disimpan di buffer yang digunakan oleh thread beban paralel untuk memuat data ke target Kafka. Nilai default adalah 50. Nilai maksimumnya adalah 1.000. Gunakan pengaturan ini dengan `ParallelLoadThreads`. `ParallelLoadBufferSize` hanya berlaku bila ada lebih dari satu thread.
- `ParallelLoadQueuesPerThread` - Gunakan pilihan ini untuk menentukan jumlah antrean yang diakses setiap thread secara bersamaan untuk mengambil catatan data dari antrean dan menghasilkan beban batch untuk target. Default adalah 1. Maksimum adalah 512.

Anda dapat meningkatkan kinerja pengambilan data perubahan (CDC) untuk titik akhir Kafka dengan menyetel pengaturan tugas untuk thread paralel dan operasi massal. Untuk melakukan ini, Anda dapat menentukan jumlah thread yang terjadi bersamaan, antrean per thread, dan jumlah catatan yang disimpan dalam buffer menggunakan pengaturan tugas `ParallelApply*`. Misalnya, Anda ingin melakukan beban CDC dan menerapkan 128 thread secara paralel. Anda juga ingin mengakses 64 antrean per thread, dengan 50 catatan disimpan per buffer.

Untuk mempromosikan kinerja CDC, AWS DMS mendukung pengaturan tugas ini:

- `ParallelApplyThreads`— Menentukan jumlah thread bersamaan yang AWS DMS digunakan selama beban CDC untuk mendorong catatan data ke titik akhir target Kafka. Nilai default adalah nol (0) dan nilai maksimum adalah 32.
- `ParallelApplyBufferSize` - Menentukan jumlah maksimum catatan yang disimpan di setiap antrean buffer untuk thread serentak untuk mendorong ke titik akhir target Kinesis selama beban CDC. Nilai default adalah 100 dan nilai maksimum adalah 1.000. Gunakan pilihan ini saat `ParallelApplyThreads` menentukan lebih dari satu thread.
- `ParallelApplyQueuesPerThread` - Menentukan jumlah antrean yang diakses oleh setiap thread untuk mengambil catatan data dari antrean dan menghasilkan beban batch untuk titik akhir Kafka selama CDC. Default-nya adalah 1. Maksimum adalah 512.

Saat menggunakan pengaturan tugas `ParallelApply*`, default `partition-key-type`-nya adalah `primary-key` dari tabel, bukan `schema-name.table-name`.

Menghubungkan ke Kafka menggunakan Keamanan Lapisan Pengangkutan (TLS)

Sebuah klaster Kafka menerima koneksi aman menggunakan Keamanan Lapisan Pengangkutan (TLS). Dengan DMS, Anda dapat menggunakan salah satu dari tiga pilihan protokol keamanan berikut untuk mengamankan koneksi titik akhir Kafka.

Enkripsi SSL (**server-encryption**)

Klien memvalidasi identitas server melalui sertifikat server. Kemudian koneksi terenkripsi dibuat antara server dan klien.

Autentikasi SSL (**mutual-authentication**)

Server dan klien memvalidasi identitas dengan satu sama lain melalui sertifikat mereka sendiri. Kemudian koneksi terenkripsi dibuat antara server dan klien.

SASL-SSL (**mutual-authentication**)

Metode The Simple Authentication and Security Layer (SASL) menggantikan sertifikat klien dengan nama pengguna dan kata sandi untuk memvalidasi identitas klien. Secara spesifik, Anda memberikan nama pengguna dan kata sandi yang telah didaftarkan ke server sehingga server dapat memvalidasi identitas klien. Kemudian koneksi terenkripsi dibuat antara server dan klien.

Important

Apache Kafka dan Amazon MSK menerima sertifikat diselesaikan. Ini adalah keterbatasan yang sudah diketahui dari Kafka dan Amazon MSK yang harus ditangani. Untuk informasi lebih lanjut, lihat [Masalah Apache Kafka, KAFKA-3700](#).

Jika Anda menggunakan Amazon MSK, pertimbangkan untuk menggunakan daftar kontrol akses (ACL) sebagai solusi untuk keterbatasan ini. Untuk informasi lebih lanjut tentang penggunaan ACL, lihat [Apache Kafka ACL](#) dalam Panduan Developer Amazon Managed Streaming for Apache Kafka.

Jika Anda menggunakan klaster Kafka yang dikelola sendiri, lihat [Komentar tanggal 21/Oct/18](#) untuk informasi tentang mengonfigurasi klaster Anda.

Menggunakan enkripsi SSL dengan Amazon MSK atau klaster Kafka yang dikelola sendiri

Anda dapat menggunakan enkripsi SSL untuk mengamankan koneksi titik akhir ke Amazon MSK atau klaster Kafka yang dikelola sendiri. Saat Anda menggunakan metode autentikasi enkripsi SSL, klien memvalidasi identitas server melalui sertifikat server. Kemudian koneksi terenkripsi dibuat antara server dan klien.

Menggunakan enkripsi SSL untuk terhubung ke Amazon MSK

- Tetapkan pengaturan titik akhir protokol keamanan (`SecurityProtocol`) menggunakan pilihan `ssl-encryption` ketika membuat titik akhir target Kafka Anda.

Contoh JSON berikut menetapkan protokol keamanan sebagai enkripsi SSL.

```
"KafkaSettings": {  
  "SecurityProtocol": "ssl-encryption",  
}
```

Untuk menggunakan enkripsi SSL sebagai klaster Kafka yang dikelola sendiri

1. Jika Anda menggunakan Otoritas Sertifikasi (CA) pribadi di klaster Kafka on-premise Anda, unggahlah sertifikat CA pribadi dan dapatkan Amazon Resource Name (ARN).
2. Tetapkan pengaturan titik akhir protokol keamanan (`SecurityProtocol`) menggunakan pilihan `ssl-encryption` ketika membuat titik akhir target Kafka Anda. Contoh JSON berikut menetapkan protokol keamanan sebagai `ssl-encryption`.

```
"KafkaSettings": {  
  "SecurityProtocol": "ssl-encryption",  
}
```

3. Jika Anda menggunakan CA pribadi, atur `SslCaCertificateArn` di ARN yang Anda dapatkan di langkah pertama di atas.

Menggunakan autentikasi SSL

Anda dapat menggunakan autentikasi SSL untuk mengamankan koneksi titik akhir ke Amazon MSK atau klaster Kafka yang dikelola sendiri.

Untuk mengaktifkan autentikasi klien dan enkripsi menggunakan otentikasi SSL untuk menyambung ke Amazon MSK, lakukan hal berikut:

- Siapkan kunci privat dan sertifikat publik untuk Kafka.
- Unggah sertifikat ke DMS Certificate Manager .
- Buat target titik akhir Kafka dengan sertifikat ARN yang sesuai yang ditentukan dalam pengaturan titik akhir Kafka.

Untuk mempersiapkan kunci privat dan sertifikat publik untuk Amazon MSK

1. Membuat instans EC2 dan atur klien untuk menggunakan autentikasi seperti yang dijelaskan dalam langkah 1 sampai 9 di [Autentikasi klien](#) Bagian dari Panduan Developer Amazon Managed Streaming for Apache Kafka.

Setelah Anda menyelesaikan langkah-langkah tersebut, Anda memiliki sertifikat-ARN (sertifikat publik ARN disimpan di ACM), dan kunci privat yang terdapat dalam file `kafka.client.keystore.jks`.

2. Dapatkan sertifikat publik dan salin sertifikat ke file `signed-certificate-from-acm.pem`, menggunakan perintah berikut:

```
aws acm-pca get-certificate --certificate-authority-arn Private_CA_ARN --  
certificate-arn Certificate_ARN
```

Perintah itu mengembalikan informasi yang serupa dengan contoh berikut:

```
{"Certificate": "123", "CertificateChain": "456"}
```

Anda kemudian menyalin yang setara dengan "123" ke file `signed-certificate-from-acm.pem`.

3. Dapatkan kunci privat dengan mengimpor kunci `msk-rsa` dari `kafka.client.keystore.jks` to `keystore.p12`, seperti yang ditunjukkan dalam contoh berikut.

```
keytool -importkeystore \  

```

```
-srckeystore kafka.client.keystore.jks \  
-destkeystore keystore.p12 \  
-deststoretype PKCS12 \  
-srcalias msk-rsa-client \  
-deststorepass test1234 \  
-destkeypass test1234
```

- Gunakan perintah berikut untuk mengekspor keystore.p12 ke format .pem.

```
openssl pkcs12 -in keystore.p12 -out encrypted-private-client-key.pem -nocerts
```

Pesan Masukkan frase PEM pass muncul dan mengidentifikasi kunci yang diterapkan untuk mengenkripsi sertifikat.

- Hapus atribut tas dan atribut kunci dari file .pem untuk memastikan bahwa baris pertama dimulai dengan string berikut.

```
---BEGIN ENCRYPTED PRIVATE KEY---
```

Untuk mengunggah sertifikat publik dan kunci privat ke DMS certificate manager dan menguji sambungan ke Amazon MSK

- Unggah ke DMS Certificate Manager menggunakan perintah berikut.

```
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
```

- Buat titik akhir target Amazon MSK dan uji koneksi untuk memastikan bahwa autentikasi TLS berfungsi.

```
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

Anda dapat menggunakan autentikasi SSL untuk mengamankan koneksi ke kluster Kafka yang dikelola sendiri. Dalam beberapa kasus, Anda mungkin menggunakan Otoritas Sertifikasi (CA) pribadi di kluster Kafka on-premise Anda. Jika demikian, unggah rantai CA Anda, sertifikat publik, dan kunci privat ke DMS certificate manager. Kemudian, gunakan Amazon Resource Name (ARN) yang sesuai dalam pengaturan titik akhir Anda ketika Anda membuat titik akhir target Kafka on-premise.

Untuk mempersiapkan kunci privat dan sertifikat yang sudah ditandatangani untuk kluster Kafka yang dikelola sendiri

1. Buat pasangan kunci seperti yang ditunjukkan pada contoh berikut.

```
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. Buat sertifikat tanda permintaan (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. Gunakan CA di truststore kluster Anda untuk menandatangani CSR. Jika Anda tidak memiliki CA, Anda dapat membuat CA pribadi Anda sendiri.

```
openssl req -new -x509 -keyout ca-key -out ca-cert -days validate-days
```

4. Impor ca-cert ke dalam server truststore dan keystore. Jika Anda tidak memiliki truststore, gunakan perintah berikut untuk membuat truststore dan impor ca-cert ke dalamnya.

```
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. Tanda tangani sertifikat.

```
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. Impor sertifikat yang sudah ditandatangani ke keystore.

```
keytool -keystore kafka.server.keystore.jks -import -file signed-certificate.pem -
alias on-premise-rsa -storepass your-keystore-password
-keypass your-key-password
```

7. Gunakan perintah berikut untuk mengimpor on-premise-rsa kunci dari kafka.server.keystore.jks ke 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. Gunakan perintah berikut untuk mengekspor keystore.p12 ke format .pem.

```
openssl pkcs12 -in keystore.p12 -out encrypted-private-server-key.pem -nocerts
```

9. Unggah encrypted-private-server-key.pem, signed-certificate.pem, dan ca-cert untuk DMS certificate manager.
10. Buat titik akhir dengan menggunakan ARN yang dikembalikan.

```
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
```

Menggunakan autentikasi SASL-SSL untuk terhubung ke Amazon MSK

Metode The Simple Authentication and Security Layer (SASL) menggunakan nama pengguna dan kata sandi untuk memvalidasi identitas klien, dan membuat koneksi terenkripsi antara server dan klien.

Untuk menggunakan SASL, pertama-tama buat nama pengguna dan kata sandi yang aman ketika mengatur Amazon MSK klaster Anda. Untuk deskripsi cara menyiapkan nama pengguna dan kata sandi yang aman untuk klaster Amazon MSK, lihat [Menyiapkan autentikasi SASL/SCRAM untuk klaster Amazon MSK](#) di Panduan Developer Amazon Managed Streaming for Apache Kafka.

Kemudian, ketika Anda membuat titik akhir target Kafka, setelah pengaturan titik akhir protokol keamanan (`SecurityProtocol`) menggunakan pilihan `sasl-ssl`. Anda juga mengatur pilihan `SaslUsername` dan `SaslPassword`. Pastikan semua ini konsisten dengan nama pengguna dan kata sandi aman yang Anda buat ketika pertama kali menyiapkan klaster Amazon MSK Anda, seperti yang ditunjukkan dalam contoh JSON berikut.

```
"KafkaSettings": {
  "SecurityProtocol": "sasl-ssl",
  "SaslUsername": "Amazon MSK cluster secure user name",
  "SaslPassword": "Amazon MSK cluster secure password"
}
```

Note

- Saat ini, hanya AWS DMS mendukung SASL-SSL yang didukung CA publik. DMS tidak mendukung SASL-SSL untuk digunakan dengan Kafka yang dikelola sendiri yang didukung oleh CA pribadi.
- Untuk otentikasi SASL-SSL, AWS DMS mendukung mekanisme SCRAM-SHA-512 secara default. AWS DMS versi 3.5.0 dan yang lebih tinggi juga mendukung mekanisme Plain. Untuk mendukung mekanisme Plain, atur `SaslMechanism` parameter tipe data `KafkaSettings` API kePLAIN.

Menggunakan citra sebelumnya untuk melihat nilai-nilai asli dari baris CDC untuk Apache Kafka sebagai target

Ketika menulis pembaruan CDC ke target data-streaming seperti Kafka Anda dapat melihat nilai asli baris basis data sumber sebelum diubah oleh pembaruan.. Untuk memungkinkan hal ini, AWS DMS mengisi gambar sebelum peristiwa pembaruan berdasarkan data yang disediakan oleh mesin database sumber.

Mesin basis data sumber yang berbeda memberikan jumlah informasi yang berbeda untuk citra sebelum:

- Oracle menyediakan update untuk kolom hanya jika mereka berubah.
- PostgreSQL menyediakan data hanya untuk kolom yang merupakan bagian dari kunci primer (berubah atau tidak). Jika replikasi logis sedang digunakan dan `REPLICA IDENTITY FULL` diatur untuk tabel sumber, Anda bisa mendapatkan seluruh informasi sebelum dan sesudah pada baris yang ditulis ke WAL dan tersedia di sini.
- MySQL umumnya menyediakan data untuk semua kolom (berubah atau tidak).

Untuk mengaktifkan pencitraan sebelumnya dengan tujuan menambahkan nilai asli dari basis data sumber untuk keluaran AWS DMS , gunakan pengaturan tugas `BeforeImageSettings` atau parameter `add-before-image-columns`. Parameter ini menerapkan aturan transformasi kolom.

`BeforeImageSettings` menambahkan atribut JSON baru untuk setiap operasi pembaruan dengan nilai yang dikumpulkan dari sistem basis data sumber, seperti yang ditunjukkan berikut.

```
"BeforeImageSettings": {
  "EnableBeforeImage": boolean,
  "FieldName": string,
  "ColumnFilter": pk-only (default) / non-lob / all (but only one)
}
```

Note

Terapkan `BeforeImageSettings` pada beban penuh ditambah tugas CDC (yang memigrasikan data yang ada dan mereplikasi perubahan yang sedang berlangsung), atau ke tugas CDC saja (yang hanya mereplikasi perubahan data). Jangan terapkan `BeforeImageSettings` pada tugas-tugas yang beban penuh saja.

Untuk pilihan `BeforeImageSettings`, hal berikut berlaku:

- Atur pilihan `EnableBeforeImage` ke `true` untuk mengaktifkan pencitraan sebelum. Default adalah `false`.
- Gunakan pilihan `FieldName` untuk memberikan nama ke atribut JSON baru. Ketika `EnableBeforeImage` adalah `true`, `FieldName` diperlukan dan tidak dapat kosong.
- Pilihan `ColumnFilter` menentukan kolom untuk menambahkan dengan menggunakan pencitraan sebelumnya. Untuk menambahkan kolom yang hanya merupakan bagian dari kunci primer tabel saja, gunakan nilai default, `pk-only`. Untuk menambahkan hanya kolom yang bukan tipe LOB, gunakan `non-lob`. Untuk menambahkan kolom apa pun yang memiliki nilai citra sebelum, gunakan `all`.

```
"BeforeImageSettings": {
  "EnableBeforeImage": true,
  "FieldName": "before-image",
  "ColumnFilter": "pk-only"
}
```

Menggunakan aturan transformasi citra sebelumnya

Sebagai alternatif untuk pengaturan tugas, Anda dapat menggunakan parameter `add-before-image-columns`, yang menerapkan aturan transformasi kolom. Dengan parameter ini, Anda dapat mengaktifkan pencitraan sebelumnya selama CDC, pada target data streaming seperti Kinesis.

Dengan menggunakan `add-before-image-columns` dalam aturan transformasi, Anda dapat menerapkan kontrol yang lebih halus dari hasil citra sebelumnya. Aturan transformasi memungkinkan Anda untuk menggunakan pencari objek yang memberikan Anda kontrol atas tabel yang dipilih untuk aturan tersebut. Dan juga, Anda dapat merangkai aturan transformasi, yang memungkinkan aturan yang berbeda untuk diterapkan ke tabel yang berbeda. Anda kemudian dapat memanipulasi kolom yang dihasilkan dengan menggunakan aturan lain.

Note

Jangan gunakan parameter `add-before-image-columns` bersamaan dengan pengaturan tugas `BeforeImageSettings` dalam tugas yang sama. Sebaliknya, gunakan salah satu saja, parameter atau pengaturan, tetapi tidak keduanya, untuk satu tugas.

Sebuah tipe peraturan `transformation` dengan parameter `add-before-image-columns` untuk kolom harus menyediakan `before-image-def` bagian. Berikut ini adalah sebuah contoh.

```
{
  "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,
  }
}
```

Nilai dari `column-prefix` ditambahkan sebelum nama kolom, dan nilai default `column-prefix` adalah `BI_`. Nilai dari `column-suffix` ditambahkan sebelum nama kolom, dan default kosong. Jangan atur `column-prefix` dan `column-suffix` ke string kosong.

Pilih satu nilai untuk `column-filter`. Untuk menambahkan kolom yang merupakan bagian dari kunci primer tabel saja, pilih `pk-only`. Untuk menambahkan kolom yang bukan tipe LOB saja, gunakan `non-lob`. Atau pilih `all` untuk menambahkan kolom yang memiliki nilai citra-sebelum.

Contoh untuk aturan transformasi citra sebelumnya

Aturan transformasi dalam contoh berikut menambahkan kolom baru yang disebut BI_emp_no pada target. Jadi pernyataan seperti UPDATE employees SET emp_no = 3 WHERE emp_no = 1; mengisi BI_emp_no field dengan 1. Ketika Anda menulis pembaruan CDC untuk target Amazon S3, kolom BI_emp_no memungkinkan untuk memberitahu baris asli yang diperbarui.

```
{
  "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"
      }
    }
  ]
}
```

Untuk informasi tentang penggunaan peraturan tindakan add-before-image-columns, lihat [Aturan dan tindakan transformasi](#).

Keterbatasan saat menggunakan Apache Kafka sebagai target AWS Database Migration Service

Batasan berikut berlaku saat menggunakan Apache Kafka sebagai target:

- AWS DMS Titik akhir target Kafka tidak mendukung kontrol akses IAM untuk Amazon Managed Streaming for Apache Kafka (Amazon MSK).
- Mode LOB penuh tidak didukung.
- Tentukan file konfigurasi Kafka untuk klaster Anda dengan properti yang memungkinkan AWS DMS untuk membuat topik baru secara otomatis. Sertakan pengaturan, `auto.create.topics.enable = true`. Jika menggunakan Amazon MSK, Anda dapat menentukan konfigurasi default ketika membuat klaster Kafka Anda, kemudian mengubah pengaturan `auto.create.topics.enable` ke `true`. Untuk informasi lebih lanjut tentang pengaturan konfigurasi default, lihat [Konfigurasi Amazon MSK default](#) di Panduan Developer Amazon Managed Streaming for Apache Kafka. Jika Anda perlu memodifikasi cluster Kafka yang ada yang dibuat menggunakan Amazon MSK, jalankan AWS CLI perintah `aws kafka create-configuration` untuk memperbarui konfigurasi Kafka Anda, seperti pada contoh berikut:

```
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"
}
```

Di sini, `~/kafka_configuration` adalah file konfigurasi yang telah Anda buat dengan pengaturan properti yang diperlukan.

Jika Anda menggunakan instans Kafka Anda sendiri yang diinstal di Amazon EC2, ubah konfigurasi cluster Kafka dengan `auto.create.topics.enable = true` pengaturan untuk AWS DMS memungkinkan Anda membuat topik baru secara otomatis, menggunakan opsi yang disediakan bersama instans Anda.

- AWS DMS menerbitkan setiap pembaruan ke satu catatan dalam database sumber sebagai satu catatan data (pesan) dalam topik Kafka tertentu terlepas dari transaksi.
- AWS DMS mendukung dua bentuk berikut untuk kunci partisi:
 - `SchemaName.TableName`: Kombinasi skema dan nama tabel.
 - `${AttributeName}`: Nilai salah satu field di JSON, atau kunci primer dari tabel dalam basis data sumber.
- `BatchApply` Tidak didukung untuk titik akhir Kafka. Menggunakan penerapan `Batch` (misalnya, pengaturan tugas metadata target `BatchApplyEnabled`) untuk target Kafka dapat mengakibatkan hilangnya data.
- AWS DMS tidak mendukung migrasi nilai tipe `BigInt` data dengan lebih dari 16 digit. Untuk mengatasi batasan ini, Anda dapat menggunakan aturan transformasi berikut untuk mengonversi `BigInt` kolom menjadi string. Untuk informasi selengkapnya tentang aturan transformasi, lihat [Aturan dan tindakan transformasi](#).

```
{
  "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
  }
}
```

Menggunakan pemetaan objek untuk bermigrasi data ke topik Kafka

AWS DMS menggunakan aturan pemetaan tabel untuk memetakan data dari sumber ke topik Kafka target. Untuk memetakan data ke topik target, Anda menggunakan jenis aturan pemetaan tabel yang disebut pemetaan objek. Anda menggunakan pemetaan objek untuk menentukan cara pencatatan data di peta sumber ke catatan data yang dipublikasikan ke topik Kafka.

topik Kafka tidak memiliki struktur preset selain memiliki kunci partisi.

Note

Anda tidak perlu menggunakan pemetaan objek. Anda dapat menggunakan pemetaan tabel biasa untuk berbagai transformasi. Namun, jenis kunci partisi akan mengikuti perilaku default ini:

- Primary Key digunakan sebagai kunci partisi untuk Full Load.
- Jika tidak ada pengaturan tugas `parallel-apply` yang digunakan, `schema.table` digunakan sebagai kunci partisi untuk CDC.
- Jika pengaturan tugas penerapan paralel digunakan, kunci utama digunakan sebagai kunci partisi untuk CDC.

Untuk membuat aturan pemetaan objek, tetapkan `rule-type` sebagai `object-mapping`. Aturan ini menentukan jenis pemetaan objek yang ingin Anda gunakan.

Struktur untuk aturan tersebut adalah sebagai berikut.

```
{
  "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 saat ini mendukung `map-record-to-record` dan `map-record-to-document` sebagai satu-satunya nilai yang valid untuk `rule-action` parameter. Pengaturan ini memengaruhi nilai yang tidak dikecualikan sebagai bagian dari daftar `exclude-columns` atribut. `map-record-to-document` Nilai `map-record-to-record` dan menentukan bagaimana AWS DMS menangani catatan ini secara default. Nilai-nilai ini tidak mempengaruhi pemetaan atribut dengan cara apapun.

Gunakan `map-record-to-record` ketika bermigrasi dari basis data relasional ke sebuah topik Kafka. Jenis aturan ini menggunakan nilai `taskResourceId.schemaName.tableName` dari basis data relasional sebagai kunci partisi dalam topik Kafka dan membuat atribut untuk masing-masing kolom dalam basis data sumber.

Saat menggunakan `map-record-to-record`, perhatikan hal berikut:

- Pengaturan ini hanya memengaruhi kolom yang dikecualikan oleh `exclude-columns` daftar.
- Untuk setiap kolom tersebut, AWS DMS buat atribut yang sesuai dalam topik target.
- AWS DMS menciptakan atribut yang sesuai ini terlepas dari apakah kolom sumber digunakan dalam pemetaan atribut.

Salah satu cara untuk memahami `map-record-to-record` adalah melihatnya beraksi. Untuk contoh ini, anggaplah bahwa Anda memulai dengan baris tabel basis data relasional dengan struktur dan data berikut.

FirstName	LastName	StoreId	HomeAddress	HomePhone	WorkAddress	WorkPhone	DateofBirth
Randy	Marsh	5	221B Baker Street	123456789 0	31 Spooner Street, Quahog	987654321 0	02/29/198 8

Untuk memigrasi informasi ini dari skema bernama `Test` ke topik Kafka, buatlah aturan untuk memetakan data ke topik target. Aturan berikut menggambarkan pemetaan.

```
{
  "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"
      }
    }
  ]
}
```

Mengingat topik Kafka dan kunci partisi (dalam hal ini, `taskResourceId.schemaName.tableName`), berikut ini menggambarkan format catatan yang dihasilkan menggunakan data sampel kami di topik target Kafka:

```
{
  "FirstName": "Randy",
  "LastName": "Marsh",
  "StoreId": "5",
  "HomeAddress": "221B Baker Street",
  "HomePhone": "1234567890",
  "WorkAddress": "31 Spooner Street, Quahog",
  "WorkPhone": "9876543210",
  "DateOfBirth": "02/29/1988"
}
```

Topik

- [Restrukturisasi data dengan pemetaan atribut](#)
- [Replikasi multitopik menggunakan pemetaan objek](#)
- [Format pesan untuk Apache Kafka](#)

Restrukturisasi data dengan pemetaan atribut

Anda dapat merestrukturisasi data saat migrasi ke topik kafka berlangsung menggunakan peta atribut. Misalnya, Anda mungkin ingin menggabungkan beberapa field di sumber menjadi field tunggal dalam target. Peta atribut berikut menggambarkan bagaimana merestrukturisasi data.

```
{
  "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}"
            }
        }
    ]
}

```

Untuk menetapkan nilai konstan untuk `partition-key`, tentukan nilai `partition-key`. Sebagai contoh, Anda mungkin melakukan ini untuk memaksa semua data untuk disimpan dalam partisi tunggal. Pemetaan berikut menggambarkan pendekatan ini.

```

{
  "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

Nilai `partition-key` untuk catatan kontrol yang ditujukan khusus untuk tabel tertentu adalah `TaskId.SchemaName.TableName`. Nilai `partition-key` untuk catatan kontrol yang ditujukan untuk tugas tertentu adalah `TaskId` dari catatan tersebut. Menentukan nilai `partition-key` dalam pemetaan objek tidak berdampak pada `partition-key` untuk catatan kontrol.

Replikasi multitopik menggunakan pemetaan objek

Secara default, AWS DMS tugas memigrasikan semua data sumber ke salah satu topik Kafka berikut:

- Seperti yang ditentukan dalam bidang `Topik` dari titik akhir AWS DMS target.
- Seperti yang ditentukan oleh `kafka-default-topic` jika bidang `Topik` dari titik akhir target tidak diisi dan `auto.create.topics.enable` pengaturan Kafka diatur ke `true`

Dengan versi AWS DMS engine 3.4.6 dan yang lebih tinggi, Anda dapat menggunakan `kafka-target-topic` atribut untuk memetakan setiap tabel sumber yang dimigrasi ke topik terpisah. Misalnya, aturan pemetaan objek berikut memigrasikan tabel sumber `Customer` dan `Address` ke topik Kafka `customer_topic` dan `address_topic`, masing-masing. Pada saat yang sama, AWS DMS memigrasikan semua tabel sumber lainnya, termasuk `Bills` tabel dalam `Test` skema, ke topik yang ditentukan di titik akhir target.

```
{
  "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"
    }
  }
]
}
```

Dengan menggunakan replikasi multitopik Kafka, Anda dapat mengelompokkan dan memigrasikan tabel sumber untuk memisahkan topik Kafka menggunakan satu tugas replikasi.

Format pesan untuk Apache Kafka

Keluaran JSON hanya berupa daftar pasangan nilai-kunci.

RecordType

Jenis catatan dapat berupa data atau kontrol. Catatan data mewakili baris yang sebenarnya dalam sumber. Kontrol catatan adalah untuk peristiwa-peristiwa penting di aliran, misalnya restart tugas.

Operasi

Untuk catatan data, operasinya dapat load, insert, update, atau delete.

Untuk catatan kontrol, operasi dapat berupa create-table, rename-table, drop-table, change-columns, add-column, drop-column, rename-column, atau column-type-change.

SchemaName

Skema sumber untuk catatan. Field ini bisa kosong untuk catatan kontrol.

TableName

Tabel sumber untuk catatan. Field ini bisa kosong untuk catatan kontrol.

Stempel waktu

Stempel waktu untuk saat pesan JSON dibuat. Field ini diformat dengan format ISO 8601.

Contoh pesan JSON berikut menggambarkan pesan tipe data dengan semua metadata tambahan.

```
{
  "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"
  }
}

```

Contoh pesan JSON berikut menggambarkan pesan tipe kontrol.

```

{
  "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"
}
```

Menggunakan AmazonOpenSearchService cluster sebagai targetAWS Database Migration Service

Anda dapat menggunakanAWS DMSuntuk memigrasi data ke AmazonOpenSearchLayanan (OpenSearchLayanan). OpenSearchLayanan adalah layanan terkelola yang membuatnya mudah untuk menyebarkan, mengoperasikan, dan menskalakanOpenSearchLayanan cluster.

DalamOpenSearchLayanan, Anda bekerja dengan indeks dan dokumen. Sebuahindeksadalah kumpulan dokumen, dandokumenadalah objek JSON yang berisi nilai-nilai skalar, array, dan benda-benda lainnya.OpenSearchmenyediakan bahasa query berbasis JSON, sehingga Anda dapat query data dalam indeks dan mengambil dokumen yang sesuai.

KapanAWS DMSmenciptakan indeks untuk endpoint target untukOpenSearchLayanan, itu menciptakan satu indeks untuk setiap tabel dari titik akhir sumber. Biaya untuk membuatOpenSearchIndeks layanan tergantung pada beberapa faktor. Ini adalah jumlah indeks yang dibuat, jumlah total data dalam indeks ini, dan sejumlah kecil metadata yangOpenSearchtoko untuk setiap dokumen.

KonfigurasiOpenSearchKlaster layanan dengan sumber daya komputasi dan penyimpanan yang sesuai untuk ruang lingkup migrasi Anda. Kami menyarankan agar Anda mempertimbangkan faktor-faktor berikut, tergantung pada tugas replikasi yang ingin Anda gunakan:

- Untuk beban data penuh, pertimbangkan jumlah total data yang ingin Anda migrasikan, dan juga kecepatan transfer.
- Untuk mereplikasi perubahan yang sedang berlangsung, pertimbangkan frekuensi pembaruan, dan persyaratan latensi end-to-end Anda.

Juga, konfigurasi pengaturan indeks padaOpenSearchcluster, memperhatikan jumlah dokumen.

Pengaturan tugas beban penuh multithread

Untuk membantu meningkatkan kecepatan transfer, AWS DMS mendukung beban penuh multithreaded ke OpenSearch Layanan target cluster. AWS DMS mendukung multithreading ini dengan pengaturan tugas yang mencakup berikut ini:

- `MaxFullLoadSubTasks` - Gunakan pilihan ini untuk menunjukkan jumlah maksimum tabel sumber yang dimuat secara paralel. DMS memuat setiap tabel ke dalam yang sesuai OpenSearch Indeks target layanan menggunakan subtugas khusus. Default adalah 8; nilai maksimum adalah 49.
- `ParallelLoadThreads` - Gunakan opsi ini untuk menentukan jumlah utas yang AWS DMS menggunakan untuk memuat setiap tabel ke dalam OpenSearch Indeks target layanan. Nilai maksimum untuk OpenSearch Target layanan adalah 32. Anda dapat meminta untuk meningkatkan batas maksimum ini.

Note

Jika Anda tidak mengubah `ParallelLoadThreads` dari default (0), AWS DMS mentransfer satu catatan pada satu waktu. Pendekatan ini menempatkan beban yang tidak semestinya pada Anda OpenSearch Layanan cluster. Pastikan bahwa Anda menetapkan pilihan ini ke 1 atau lebih.

- `ParallelLoadBufferSize` - Gunakan opsi ini untuk menentukan jumlah maksimum catatan untuk disimpan dalam buffer yang digunakan benang beban paralel untuk memuat data ke OpenSearch Target layanan. Nilai default adalah 50. Nilai maksimumnya adalah 1.000. Gunakan pengaturan ini dengan `ParallelLoadThreads`. `ParallelLoadBufferSize` hanya berlaku bila ada lebih dari satu thread.

Untuk informasi lebih lanjut tentang bagaimana DMS memuat OpenSearch Layanan cluster menggunakan multithreading, lihat AWS posting blog [Skala Amazon OpenSearch Layanan untuk AWS Database Migration Service migrasi](#).

Pengaturan beban tugas CDC multithread

Anda dapat meningkatkan kinerja pengambilan data perubahan (CDC) untuk OpenSearch Layanan target cluster menggunakan pengaturan tugas untuk memodifikasi perilaku `PutRecords` Panggilan API. Untuk melakukan ini, Anda dapat menentukan jumlah thread yang terjadi bersamaan, antrian per thread, dan jumlah catatan untuk disimpan dalam buffer menggunakan pengaturan tugas

`ParallelApply*`. Misalnya, Anda ingin melakukan beban CDC dan menerapkan 32 thread secara paralel. Anda juga ingin mengakses 64 antrian per thread, dengan 50 catatan disimpan per buffer.

Note

Dukungan untuk penggunaan `ParallelApply*` pengaturan tugas selama CDC ke `AmazonOpenSearchEndpoint` target layanan tersedia di `AWS DMS` versi 3.4.0 dan yang lebih tinggi

Untuk mempromosikan performa CDC, `AWS DMS` mendukung pengaturan tugas ini:

- `ParallelApplyThreads`- Menentukan jumlah benang bersamaan yang `AWS DMS` menggunakan selama beban CDC untuk mendorong catatan data ke `OpenSearch` titik akhir target layanan. Nilai default adalah nol (0) dan nilai maksimum adalah 32.
- `ParallelApplyBufferSize`- Menentukan jumlah maksimum catatan untuk menyimpan di setiap antrian buffer untuk benang bersamaan untuk mendorong ke `OpenSearch` titik akhir target layanan selama beban CDC. Nilai default adalah 100 dan nilai maksimum adalah 1.000. Gunakan pilihan ini saat `ParallelApplyThreads` menentukan lebih dari satu thread.
- `ParallelApplyQueuesPerThread`- Menentukan jumlah antrian yang setiap thread mengakses untuk mengambil catatan data dari antrian dan menghasilkan beban batch untuk `OpenSearch` titik akhir layanan selama CDC.

Saat menggunakan pengaturan tugas `ParallelApply*`, default `partition-key-type`-nya adalah `primary-key` dari tabel, bukan `schema-name.table-name`.

Migrasi dari tabel database relasional ke `OpenSearch` Indeks layanan

`AWS DMS` mendukung migrasi data ke `OpenSearch` jenis data skalar layanan. Saat bermigrasi dari database relasional seperti Oracle atau MySQL ke `OpenSearch` layanan, Anda mungkin ingin merestrukturisasi bagaimana Anda menyimpan data ini.

`AWS DMS` mendukung berikut `OpenSearch` jenis data skalar layanan:

- Boolean
- Date
- Float
- Int

- String

AWS DMS mengonversi data dari tipe Date ke tipe String. Anda dapat menentukan pemetaan kustom untuk menafsirkan tanggal ini.

AWS DMS tidak mendukung migrasi tipe data LOB.

Prasyarat untuk menggunakan AmazonOpenSearchLayanan sebagai targetAWS Database Migration Service

Sebelum Anda mulai bekerja denganOpenSearchDatabase layanan sebagai targetAWS DMS, pastikan bahwa Anda membuatAWS Identity and Access Management(IAM) peran. Peran ini harus membiarkanAWS DMSmengaksesOpenSearchIndeks layanan pada titik akhir target. Kumpulan izin akses minimum ditampilkan dalam kebijakan IAM berikut.

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

Peran yang Anda gunakan untuk migrasiOpenSearchLayanan harus memiliki izin berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttpDelete",

```

```

        "es:ESHttpGet",
        "es:ESHttpGet",
        "es:ESHttpPost",
        "es:ESHttpPut"
    ],
    "Resource": "arn:aws:es:region:account-id:domain/domain-name/*"
}
]
}

```

Pada contoh sebelumnya, ganti *region* dengan AWS Pengenal wilayah, *account-id* dengan Anda AWS ID akun, dan *domain-name* dengan nama Amazon Anda OpenSearch Domain layanan. Contohnya `arn:aws:es:us-west-2:123456789012:domain/my-es-domain`

Pengaturan titik akhir saat menggunakan OpenSearch Layanan sebagai target AWS DMS

Anda dapat menggunakan pengaturan titik akhir untuk mengonfigurasi OpenSearch Target layanan database mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir target menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan `--elasticsearch-settings '{"EndpointSetting": "value", ...}'` sintaks JSON.

Tabel berikut menunjukkan pengaturan titik akhir yang dapat Anda gunakan OpenSearch Layanan sebagai target.

Nama atribut	Nilai valid	Nilai dan deskripsi default
<code>FullLoadErrorPercentage</code>	Sebuah integer positif lebih besar dari 0 tetapi tidak lebih besar dari 100.	10 - Untuk tugas beban penuh, atribut ini menentukan ambang batas kesalahan yang diizinkan sebelum tugas gagal. Sebagai contoh, ada 1.500 baris di titik akhir sumber dan parameter ini diatur ke 10. Kemudian tugas gagal jika AWS DMS menemukan lebih dari 150 kesalahan (10 persen dari jumlah baris) saat menulis ke titik akhir target.
<code>ErrorRetryDuration</code>	Sebuah integer positif lebih besar dari 0.	300 - Jika terjadi kesalahan pada titik akhir target, AWS DMS mencoba lagi selama beberapa detik ini. Jika tidak, tugas gagal.

Batasan saat menggunakan AmazonOpenSearchLayanan sebagai targetAWS Database Migration Service

Batasan berikut berlaku saat menggunakan AmazonOpenSearchLayanan sebagai target:

- OpenSearchLayanan menggunakan pemetaan dinamis (auto guess) untuk menentukan tipe data yang akan digunakan untuk data yang dimigrasi.
- OpenSearchLayanan menyimpan setiap dokumen dengan ID unik. Berikut adalah contoh ID.

```
"_id": "D359F8B537F1888BC71FE20B3D79EAE6674BE7ACA9B645B0279C7015F6FF19FD"
```

Setiap ID dokumen adalah panjangnya 64 byte, jadi antisipasi ini sebagai syarat penyimpanan. Misalnya, jika Anda memigrasi 100.000 baris dariAWS DMSsumber, yang dihasilkanOpenSearchIndeks layanan memerlukan penyimpanan untuk tambahan 6.400.000 byte.

- DenganOpenSearchLayanan, Anda tidak dapat membuat pembaruan untuk atribut kunci utama. Pembatasan ini penting ketika menggunakan replikasi berkelanjutan dengan perubahan pengambilan data (CDC) karena dapat mengakibatkan data yang tidak diinginkan dalam target. Dalam mode CDC, kunci primer dipetakan ke nilai SHA256, dengan panjang 32 byte. Ini diubah menjadi string 64-byte yang dapat dibaca manusia, dan digunakan sebagaiOpenSearchID dokumen layanan.
- JikaAWS DMSmenemukan item yang tidak dapat dimigrasi, ia menulis pesan kesalahan ke AmazonCloudWatchLog. Perilaku ini berbeda dari titik akhir target AWS DMS lainnya, yang menulis kesalahan ke tabel pengecualian.
- AWS DMS tidak mendukung koneksi ke kluster Amazon ES yang memiliki Fine-grained Access Control diaktifkan dengan pengguna master dan kata sandi.
- AWS DMStidak mendukungOpenSearchLayanan tanpa server.
- OpenSearchLayanan tidak mendukung penulisan data ke indeks yang sudah ada sebelumnya.

Jenis data target untuk AmazonOpenSearchLayanan

Saat AWS DMS memigrasikan data dari basis data heterogen, layanan memetakan jenis data dari basis data sumber ke tipe data perantara yang disebut tipe data AWS DMS. Layanan kemudian memetakan tipe data perantara ke tipe data target. Tabel berikut menunjukkan masing-masingAWS DMS tipe data dan tipe data yang dipetakan ke dalamOpenSearchLayanan.

Tipe data AWS DMS	OpenSearchTipe data layanan
Boolean	boolean
Date	string
Time	date
Timestamp	date
INT4	Integer
Real4	float
UINT4	Integer

Untuk informasi tambahan tentang tipe data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Menggunakan Amazon DocumentDB sebagai target untuk Database Migration Service AWS

Untuk informasi tentang versi Amazon DocumentDB (dengan kompatibilitas MongoDB) AWS DMS mendukung, lihat [Target untuk AWS DMS](#). Anda dapat menggunakan AWS DMS untuk memigrasikan data ke Amazon DocumentDB (dengan kompatibilitas MongoDB) dari salah satu mesin data sumber yang didukung AWS DMS. Mesin sumber dapat berada di AWS layanan terkelola seperti Amazon RDS, Aurora, atau Amazon S3. Atau mesin dapat berada di basis data terkelola mandiri, seperti MongoDB yang berjalan di Amazon EC2 atau on-premise.

Anda dapat menggunakan AWS DMS untuk mereplikasi data sumber ke basis data, koleksi, atau dokumen Amazon DocumentDB.

Note

Jika titik akhir sumber adalah MongoDB atau Amazon DocumentDB, jalankan migrasi dalam Mode dokumen.

MongoDB menyimpan data dalam format binary JSON (BSON). AWS DMS mendukung semua tipe data BSON yang didukung oleh Amazon DocumentDB. Untuk daftar tipe data ini, lihat [API, operasi, dan tipe data yang didukung MongoDB](#) dalam Panduan Developer Amazon DocumentDB.

Jika titik akhir sumber adalah basis data relasional, AWS DMS memetakan objek basis data untuk Amazon DocumentDB sebagai berikut:

- Sebuah basis data relasional, atau skema basis data, dipetakan ke Amazon DocumentDB basis data.
- Tabel di dalam peta basis data relasional dipetakan ke Koleksi di Amazon DocumentDB.
- Catatan dalam peta tabel relasional dipetakan ke Dokumen di Amazon DocumentDB. Setiap dokumen disusun dari data dalam catatan sumber.

Jika titik akhir sumber adalah Amazon S3, maka objek Amazon DocumentDB yang dihasilkan sesuai dengan AWS DMS aturan pemetaan untuk Amazon S3. Contohnya, perhatikan URI berikut ini.

```
s3://mybucket/hr/employee
```

Dalam kasus ini, AWS DMS memetakan objek di mybucket ke Amazon DocumentDB sebagai berikut:

- Bagian URI tingkat atas (hr) dipetakan ke basis data Amazon DocumentDB.
- Bagian URI berikutnya (employee) dipetakan ke koleksi Amazon DocumentDB.
- Setiap objek dalam employee dipetakan ke dokumen di Amazon DocumentDB.

Untuk informasi lebih lanjut tentang aturan pemetaan untuk Amazon S3, lihat [Menggunakan Amazon S3 sebagai sumber untuk AWS DMS](#).

Pengaturan titik akhir Amazon DocumentDB

Di AWS DMS versi 3.5.0 dan lebih tinggi, Anda dapat meningkatkan performa perubahan pengambilan data (CDC) untuk titik akhir Amazon DocumentDB dengan menggunakan pengaturan tugas untuk utas paralel dan operasi massal. Untuk melakukan ini, Anda dapat menentukan jumlah thread yang terjadi bersamaan, antrean per thread, dan jumlah catatan untuk disimpan dalam buffer menggunakan pengaturan tugas `ParallelApply*`. Misalnya, Anda ingin melakukan beban CDC dan menerapkan 128 thread secara paralel. Anda juga ingin mengakses 64 antrean per thread, dengan 50 catatan disimpan per buffer.

Untuk mempromosikan performa CDC, AWS DMS mendukung pengaturan tugas ini:

- `ParallelApplyThreads`- Menentukan jumlah thread secara langsung AWS DMS digunakan selama pemuatan CDC untuk mendorong catatan data ke titik akhir target Amazon DocumentDB. Nilai default adalah nol (0) dan nilai maksimum adalah 32.
- `ParallelApplyBufferSize`— Menentukan jumlah maksimum catatan untuk disimpan di setiap antrean buffer untuk utas secara langsung untuk mendorong ke titik akhir target Amazon DocumentDB selama pemuatan CDC. Nilai default adalah 100 dan nilai maksimum adalah 1.000. Gunakan pilihan ini saat `ParallelApplyThreads` menentukan lebih dari satu thread.
- `ParallelApplyQueuesPerThread`— Menentukan jumlah antrian yang diakses setiap utas untuk mengeluarkan data dari antrian dan menghasilkan pemuatan batch untuk titik akhir Amazon DocumentDB selama CDC. Default-nya adalah 1. Maksimal adalah 512.

Untuk detail tambahan tentang bekerja dengan Amazon DocumentDB sebagai target untuk AWS DMS, lihat bagian berikut:

Topik

- [Pemetaan data dari sumber ke target Amazon DocumentDB](#)
- [Menghubungkan ke Amazon DocumentDB Elastic Clusters sebagai target](#)
- [Replikasi berkelanjutan dengan Amazon DocumentDB sebagai target](#)
- [Keterbatasan penggunaan Amazon DocumentDB sebagai target](#)
- [Menggunakan pengaturan titik akhir dengan Amazon DocumentDB sebagai target](#)
- [Target tipe data untuk Amazon DocumentDB](#)

 Note

Untuk step-by-step panduan proses migrasi, lihat [Migrasi dari MongoDB ke Amazon DocumentDB](#) di AWS Database Migration Service Panduan Migrasi Langkah-demi-langkah.

Pemetaan data dari sumber ke target Amazon DocumentDB

AWS DMS membaca catatan dari titik akhir sumber, dan membuat dokumen JSON berdasarkan data yang dibaca. Untuk setiap dokumen JSON, AWS DMS harus menentukan field `_id` untuk bertindak

sebagai pengenal unik. Kemudian menulis dokumen JSON ke koleksi Amazon DocumentDB, menggunakan `_id` sebagai kunci primer.

Sumber data yang berupa kolom tunggal

Jika sumber data terdiri dari satu kolom, data harus bertipe string. (Tergantung pada mesin sumber, tipe data sebenarnya mungkin VARCHAR, NVARCHAR, TEXT, LOB, CLOB, atau yang sejenis.) AWS DMS mengasumsikan bahwa data adalah dokumen JSON yang valid, dan mereplikasi data ke Amazon DocumentDB apa adanya.

Jika dokumen JSON yang dihasilkan berisi field bernama `_id`, maka field tersebut digunakan sebagai unik `_id` di Amazon DocumentDB.

Jika JSON tidak berisi field `_id`, maka Amazon DocumentDB menghasilkan nilai `_id` secara otomatis.

Sumber data yang terdiri dari beberapa kolom

Jika sumber data terdiri dari beberapa kolom, maka AWS DMS membangun dokumen JSON dari semua kolom ini. Untuk menentukan field `_id` untuk dokumen, AWS DMS melakukan sebagai berikut:

- Jika salah satu kolom bernama `_id`, maka data dalam kolom tersebut yang digunakan sebagai target `_id`.
- Jika tidak ada kolom `_id`, tetapi sumber data memiliki kunci primer atau indeks unik, maka AWS DMS menggunakan kunci atau indeks nilai tersebut sebagai nilai `_id`. Data dari kunci primer atau indeks unik juga muncul sebagai field eksplisit dalam dokumen JSON.
- Jika tidak ada kolom `_id`, dan tidak ada kunci primer atau indeks unik, maka Amazon DocumentDB menghasilkan nilai `_id` secara otomatis.

Memaksakan tipe data pada titik akhir target

AWS DMS dapat memodifikasi struktur data ketika menulis ke titik akhir target Amazon DocumentDB. Anda dapat meminta perubahan ini dengan mengubah nama kolom dan tabel di titik akhir sumber, atau dengan menyediakan aturan transformasi yang diterapkan ketika tugas berjalan.

Menggunakan dokumen JSON nested (json_prefix)

Untuk memaksakan tipe data, Anda dapat mengawali nama kolom sumber dengan `json_` (yaitu, `json_columnName`) baik secara manual atau menggunakan transformasi. Dalam kasus ini, kolom dibuat sebagai dokumen JSON nested dalam dokumen target, bukan sebagai field string.

Misalnya, anggaplah Anda ingin memigrasikan dokumen berikut dari titik akhir sumber MongoDB.

```
{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",
  "ContactDetails": "{\"Home\": {\"Address\": \"Boston\", \"Phone\": \"1111111\"}, \"Work\": { \"Address\": \"Boston\", \"Phone\": \"2222222222\"}}"
```

Jika Anda tidak memaksa salah satu tipe sumber data, dokumen `ContactDetails` yang tertanam akan dimigrasi sebagai string.

```
{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",
  "ContactDetails": "{\"Home\": {\"Address\": \"Boston\", \"Phone\": \"1111111\"}, \"Work\": { \"Address\": \"Boston\", \"Phone\": \"2222222222\"}}"
```

Namun, Anda dapat menambahkan aturan transformasi untuk memaksa `ContactDetails` menjadi objek JSON. Misalnya, anggaplah bahwa nama kolom sumber asli adalah `ContactDetails`. Untuk memaksa tipe data sebagai Nested JSON, kolom di titik akhir sumber perlu diganti namanya menjadi `json_ContactDetails` baik dengan menambahkan awalan `*json_*` pada sumber secara manual atau melalui aturan transformasi. Misalnya, Anda dapat menggunakan aturan transformasi di bawah ini:

```
{
  "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 mereplikasi ContactDetails bidang sebagai JSON bersarang, sebagai berikut.

```

{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",
  "ContactDetails": {
    "Home": {
      "Address": "Boston",
      "Phone": "1111111111"
    },
    "Work": {
      "Address": "Boston",
      "Phone": "2222222222"
    }
  }
}

```

Menggunakan array JSON (array_prefix)

Untuk memaksakan tipe data, Anda dapat mengawali nama kolom dengan `array_` (yaitu, `array_columnName`), baik secara manual maupun menggunakan transformasi. Dalam kasus ini, AWS DMS menganggap kolom sebagai array JSON, dan membuatnya seperti itu dalam dokumen target.

Misalnya Anda ingin memigrasikan dokumen berikut dari titik akhir sumber MongoDB.

```

{
  "_id" : "1",
  "FirstName": "John",
  "LastName": "Doe",

```

```

"ContactAddresses": ["Boston", "New York"],
"ContactPhoneNumbers": ["1111111111", "2222222222"]
}

```

Jika Anda tidak memaksakan salah satu tipe sumber data, dokumen `ContactDetails` yang tertanam akan dimigrasi sebagai string.

```

{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",

  "ContactAddresses": "[\"Boston\", \"New York\"]",
  "ContactPhoneNumbers": "[\"1111111111\", \"2222222222\"]"
}

```

Namun, Anda dapat menambahkan aturan transformasi untuk memaksa `ContactAddress` dan `ContactPhoneNumbers` ke array JSON, seperti yang ditunjukkan pada tabel berikut.

Nama kolom sumber asli	Kolom sumber berganti nama
<code>ContactAddress</code>	<code>array_ContactAddress</code>
<code>ContactPhoneNumbers</code>	<code>array_ContactPhoneNumbers</code>

AWS DMS mereplikasi `ContactAddress` dan `ContactPhoneNumbers` sebagai berikut.

```

{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",
  "ContactAddresses": [
    "Boston",
    "New York"
  ],
  "ContactPhoneNumbers": [
    "1111111111",

```

```
"2222222222"  
  ]  
}
```

Menghubungkan ke Amazon DocumentDB menggunakan TLS

Secara default, klaster Amazon DocumentDB yang baru dibuat hanya menerima sambungan aman menggunakan Keamanan Lapisan Pengangkutan (TLS). Ketika TLS diaktifkan, setiap sambungan ke Amazon DocumentDB memerlukan kunci publik.

Anda dapat mengambil kunci publik untuk Amazon DocumentDB dengan mengunduh file `rds-combined-ca-bundle.pem`, dari sebuah AWS S3. Untuk informasi lebih lanjut tentang mengunduh file ini, lihat [Mengkripsi sambungan menggunakan TLS](#) dalam Panduan Developer Amazon DocumentDB

Setelah Anda mengunduh file, Anda dapat mengimpor kunci publik yang ada di dalamnya ke AWS DMS seperti dijelaskan berikut ini.

AWS Management Console

Untuk mengimpor file kunci publik (.pem)

1. Buka konsol AWS DMS di <https://console.aws.amazon.com/dms>.
2. Pilih Sertifikat di panel navigasi.
3. Pilih Impor Sertifikat dan lakukan hal berikut:
 - Untuk Pengenal sertifikat, masukkan nama unik untuk sertifikat tersebut, contohnya `docdb-cert`.
 - Untuk Impor file, arahkan ke lokasi tempat Anda menyimpan file .pem.

Jika pengaturan sudah sesuai keinginan Anda, pilih Tambahkan Sertifikat CA baru.

AWS CLI

Gunakan perintah `aws dms import-certificate` seperti pada contoh berikut.

```
aws dms import-certificate \  
  --certificate-identifier docdb-cert \  
  --certificate-pem file:///./rds-combined-ca-bundle.pem
```

Saat Anda membuat titik akhir target AWS DMS, berikan pengidentifikasi sertifikat (misalnya, docdb-cert). Juga, tetapkan parameter mode SSL ke `verify-full`.

Menghubungkan ke Amazon DocumentDB Elastic Clusters sebagai target

Di AWS DMS versi 3.4.7 dan yang lebih tinggi, Anda dapat membuat titik akhir target Amazon DocumentDB sebagai Elastic Cluster. Jika Anda membuat titik akhir target sebagai Elastic Cluster, Anda harus melampirkan sertifikat SSL baru ke titik akhir Amazon DocumentDB Elastic Cluster karena sertifikat SSL yang ada tidak akan berfungsi.

Untuk melampirkan sertifikat SSL baru ke titik akhir Amazon DocumentDB Elastic Cluster

1. Di browser, buka <https://www.amazontrust.com/repository/SFSRootCAG2.pem> dan menyimpan isinya ke `.pem` file dengan nama file yang unik, misalnya `SFSRootCAG2.pem`. Ini adalah file sertifikat yang perlu Anda impor pada langkah selanjutnya.
2. Buat titik akhir Cluster Elastis dan atur opsi berikut:
 - a. Di bawah Konfigurasi titik akhir, pilih **Tambahkan sertifikat CA baru**.
 - b. Untuk Pengidentifikasi sertifikat, masukkan **SFSRootCAG2.pem**.
 - c. Untuk Impor file sertifikat, pilih **Pilih berkas**, lalu arahkan ke `SFSRootCAG2.pem` file yang sebelumnya Anda unduh.
 - d. Pilih dan buka yang diunduh `SFSRootCAG2.pem` berkas.
 - e. Pilih **Impor sertifikat**.
 - f. Dari **Pilih sertifikat** drop down, pilih `SFSRootCag2.pem`.

Sertifikat SSL baru dari yang diunduh `SFSRootCAG2.pem` file sekarang dilampirkan ke titik akhir Amazon DocumentDB Elastic Cluster Anda.

Replikasi berkelanjutan dengan Amazon DocumentDB sebagai target

Jika replikasi yang sedang berlangsung (ubah pengambilan data, CDC) diaktifkan untuk Amazon DocumentDB sebagai target, AWS DMS versi 3.5.0 dan lebih tinggi memberikan peningkatan kinerja yang dua puluh kali lebih besar daripada rilis sebelumnya. Dalam rilis sebelumnya di mana AWS DMS menangani hingga 250 catatan per detik, AWS DMS sekarang menangani sekitar 5000 rekor/detik. AWS DMS juga memastikan bahwa dokumen di Amazon DocumentDB tetap sinkron dengan sumbernya. Ketika catatan sumber dibuat atau diperbarui, AWS DMS pertama-tama harus menentukan catatan Amazon DocumentDB mana yang terpengaruh dengan melakukan hal berikut:

- Jika catatan sumber memiliki kolom bernama `_id`, nilai kolom tersebut menentukan `_id` yang sesuai dalam koleksi Amazon DocumentDB.
- Jika tidak ada kolom `_id`, tetapi sumber data memiliki kunci primer atau indeks unik, maka AWS DMS menggunakan kunci atau nilai indeks tersebut sebagai `_id` untuk koleksi Amazon DocumentDB.
- Jika catatan sumber tidak memiliki kolom `_id`, kunci primer, maupun indeks unik, maka AWS DMS mencocokkan semua kolom sumber ke field yang sesuai dalam koleksi Amazon DocumentDB.

Saat catatan sumber baru dibuat, AWS DMS menulis dokumen terkait ke Amazon DocumentDB. Jika catatan sumber yang ada diperbarui, AWS DMS memperbarui field terkait dalam dokumen target di Amazon DocumentDB. Setiap field yang ada di dokumen target tetapi tidak ada di catatan sumber tetap tak tersentuh.

Ketika sebuah catatan sumber dihapus, AWS DMS menghapus dokumen terkait dari Amazon DocumentDB.

Perubahan struktural (DDL) pada sumbernya

Dengan replikasi berkelanjutan, setiap perubahan pada struktur data sumber (seperti tabel, kolom, dan sebagainya) disebarkan ke rekan-rekan mereka di Amazon DocumentDB. Dalam basis data relasional, perubahan ini dimulai dengan menggunakan pernyataan bahasa definisi data (DDL). Anda dapat melihat bagaimana AWS DMS menyebarkan perubahan ini ke Amazon DocumentDB dalam tabel berikut.

DDL di sumber	Efek pada target Amazon DocumentDB
CREATE TABLE	Membuat koleksi kosong.
Pernyataan yang mengganti nama tabel (RENAME TABLE, ALTER TABLE...RENAME, dan sejenisnya)	Mengganti nama koleksi.
TRUNCATE TABLE	Menghapus semua dokumen dari koleksi, tetapi hanya jika <code>HandleSourceTableTruncated</code> adalah <code>true</code> . Untuk informasi lebih lanjut, lihat Pengaturan tugas untuk perubahan pengolahan DDL penanganan .

DDL di sumber	Efek pada target Amazon DocumentDB
DROP TABLE	Menghapus koleksi, tetapi hanya jika <code>HandleSourceTableDropped</code> adalah <code>true</code> . Untuk informasi lebih lanjut, lihat Pengaturan tugas untuk perubahan pengolahan DDL penanganan .
Pernyataan yang menambahkan kolom ke tabel (ALTER TABLE...ADD dan sejenisnya)	Pernyataan DDL diabaikan, dan peringatan dikeluarkan. Ketika INSERT pertama kali dilakukan pada sumber, field baru ditambahkan ke dokumen target.
ALTER TABLE...RENAME COLUMN	Pernyataan DDL diabaikan, dan peringatan dikeluarkan. Ketika INSERT pertama kali dilakukan pada sumber, field baru ditambahkan ke dokumen target.
ALTER TABLE...DROP COLUMN	Pernyataan DDL diabaikan, dan peringatan dikeluarkan.
Pernyataan yang mengubah tipe data kolom (ALTER COLUMN...MODIFY dan sejenisnya)	Pernyataan DDL diabaikan, dan peringatan dikeluarkan. Ketika INSERT pertama kali dilakukan pada sumber dengan tipe data baru, dokumen target dibuat dengan field tipe data baru tersebut.

Keterbatasan penggunaan Amazon DocumentDB sebagai target

Batasan berikut berlaku saat menggunakan Amazon DocumentDB sebagai target untuk AWS DMS:

- Di Amazon DocumentDB, nama koleksi tidak dapat berisi simbol dolar (\$). Selain itu, nama basis data tidak dapat berisi karakter Unicode.
- AWS DMS tidak mendukung penggabungan beberapa tabel sumber menjadi koleksi Amazon DocumentDB tunggal.
- Saat AWS DMS memproses perubahan dari tabel sumber yang tidak memiliki kunci primer, kolom LOB apa pun dalam tabel tersebut diabaikan.

- Jika pilih Ubah tabel diaktifkan dan AWS DMS menemukan kolom sumber bernama "_id", maka kolom tersebut muncul sebagai "__id" (dua garis bawah) di tabel perubahan.
- Jika Anda memilih Oracle sebagai titik akhir sumber, maka sumber Oracle harus memiliki pencatatan tambahan penuh diaktifkan. Jika tidak, ketika ada kolom pada sumber yang tidak berubah, maka data dimuat ke Amazon DocumentDB sebagai nilai null.
- Pengaturan tugas replikasi, `TargetTablePrepMode: TRUNCATE_BEFORE_LOAD` tidak didukung untuk digunakan dengan titik akhir target DocumentDB.

Menggunakan pengaturan titik akhir dengan Amazon DocumentDB sebagai target

Anda dapat menggunakan pengaturan titik akhir untuk mengonfigurasi basis data target Amazon DocumentDB mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat membuat titik akhir target menggunakan AWS DMS dengan menggunakan konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan `--doc-db-settings '{"EndpointSetting": "value", ...}'` Sintaks JSON.

Tabel berikut menunjukkan pengaturan titik akhir yang dapat Anda gunakan dengan Amazon DocumentDB sebagai target.

Nama atribut	Nilai valid	Nilai dan deskripsi default
<code>replicateShardCollections</code>	boolean true false	<p>Kapan <code>true</code>, pengaturan titik akhir ini memiliki efek berikut dan memberlakukan batasan berikut:</p> <ul style="list-style-type: none"> • AWS DMS diizinkan untuk mereplikasi data untuk menargetkan koleksi pecahan. Pengaturan ini hanya berlaku jika titik akhir DocumentDB target adalah Elastic Cluster. • Anda harus mengatur <code>TargetTablePrepMode</code> kepada <code>DO_NOTHING</code>. • AWS DMS secara otomatis mengatur <code>useUpdateLookup</code> kepada <code>false</code> selama migrasi.

Target tipe data untuk Amazon DocumentDB

Dalam tabel berikut, Anda dapat menemukan tipe data target Amazon DocumentDB yang didukung saat menggunakan AWS DMS, dan pemetaan default dari AWS tipe data DMS. Untuk informasi lebih lanjut tentang tipe data DMS AWS, lihat [Tipe data untuk AWS Database Migration Service](#).

Tipe data DMS AWS	Tipe data Amazon DocumentDB
BOOLEAN	Boolean
BYTES	Binary data
DATE	Date
TIME	String (UTF8)
DATETIME	Date
INT1	32-bit integer
INT2	32-bit integer
INT4	32-bit integer
INT8	64-bit integer
NUMERIC	String (UTF8)
REAL4	Double
REAL8	Double
STRING	Jika data dikenali sebagai JSON, maka AWS DMS memigrasi kannya ke Amazon DocumentDB sebagai dokumen. Jika tidak, data tersebut dipetakan ke String (UTF8).
UINT1	32-bit integer
UINT2	32-bit integer
UINT4	64-bit integer

Tipe data DMS AWS	Tipe data Amazon DocumentDB
UINT8	String (UTF8)
WSTRING	Jika data dikenali sebagai JSON, maka AWS DMS memigrasi kannya ke Amazon DocumentDB sebagai dokumen. Jika tidak, data tersebut dipetakan ke String (UTF8).
BLOB	Binary
CLOB	Jika data dikenali sebagai JSON, maka AWS DMS memigrasi kannya ke Amazon DocumentDB sebagai dokumen. Jika tidak, data tersebut dipetakan ke String (UTF8).
NCLOB	Jika data dikenali sebagai JSON, maka AWS DMS memigrasi kannya ke Amazon DocumentDB sebagai dokumen. Jika tidak, data tersebut dipetakan ke String (UTF8).

Menggunakan Amazon Neptune sebagai target untuk AWS Database Migration Service

Amazon Neptune adalah layanan basis data grafik yang cepat, andal, terkelola sepenuhnya yang membuatnya mudah dalam membangun dan menjalankan aplikasi yang bekerja dengan set data yang sangat terhubung. Inti dari Neptune adalah mesin basis data grafik berperforma tinggi yang dibuat khusus. Mesin ini dioptimalkan untuk menyimpan miliaran hubungan dan membuat kueri grafik dengan latensi milidetik. Neptune mendukung bahasa query grafik populer Apache TinkerPop Gremlin dan SPARQL W3C. Untuk informasi lebih lanjut tentang Amazon Neptune, lihat [Apa itu Amazon Neptune?](#) di Panduan Pengguna Amazon Neptune.

Tanpa basis data grafik seperti Neptune, Anda mungkin memodelkan data yang sangat terhubung dalam basis data relasional. Karena data memiliki koneksi dengan berpotensi dinamis, aplikasi yang menggunakan sumber data tersebut harus memodelkan kueri data terhubung di SQL. Pendekatan ini mengharuskan Anda menulis sebuah lapisan tambahan untuk mengonversi kueri grafik ke SQL. Dan juga, basis data relasional tersedia dengan skema kekakuan. Setiap perubahan dalam skema untuk memodelkan koneksi yang berubah memerlukan downtime dan pemeliharaan tambahan dari konversi kueri untuk mendukung skema baru. Performa kueri juga merupakan hambatan besar lain yang harus dipertimbangkan saat merancang aplikasi Anda.

Basis data grafik dapat menyederhanakan situasi seperti itu secara signifikan. Tidak terikat skema, lapisan kueri yang kaya akan grafik (Gremlin atau SPARQL) dan indeks yang dioptimalkan untuk kueri grafik akan meningkatkan fleksibilitas dan performa. Basis data grafik Amazon Neptune juga memiliki fitur untuk korporasi seperti encryption at rest, lapisan otorisasi aman, backup default, dukungan Multi-AZ, support replika baca, dan lain-lain.

Menggunakan AWS DMS, Anda dapat memigrasi data relasional yang memodelkan grafik yang sangat terhubung ke titik akhir target Neptune dari titik akhir sumber DMS untuk basis data SQL yang didukung.

Untuk detailnya, lihat berikut ini.

Topik

- [Gambaran umum tentang migrasi ke Amazon Neptune sebagai target](#)
- [Menentukan pengaturan titik akhir untuk Amazon Neptune sebagai target](#)
- [Membuat peran layanan IAM untuk mengakses Amazon Neptune sebagai target](#)
- [Menentukan aturan pemetaan grafik menggunakan Gremlin dan R2RLL untuk Amazon Neptune sebagai target](#)
- [Tipe data untuk migrasi Gremlin dan R2RLL ke Amazon Neptune sebagai target](#)
- [Keterbatasan menggunakan Amazon Neptune sebagai target](#)

Gambaran umum tentang migrasi ke Amazon Neptune sebagai target

Sebelum memulai migrasi ke target Neptune, buat sumber daya berikut di akun AWS Anda:

- Sebuah klaster Neptune untuk titik akhir target.
- Sebuah basis data relasional SQL didukung oleh AWS DMS untuk titik akhir sumber.
- Bucket Amazon S3 untuk titik akhir target. Buat bucket S3 ini di Wilayah AWS yang sama dengan klaster Neptune Anda. AWS DMS menggunakan bucket S3 ini sebagai penyimpanan file perantara untuk data target yang sebagian besar dimuat ke basis data Neptune. Untuk informasi selengkapnya tentang cara membuat bucket S3, lihat [Membuat ember](#) di dalam Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- titik akhir virtual private cloud (VPC) untuk S3 di VPC yang sama seperti klaster Neptune.
- Sebuah AWS Identity and Access Management (IAM) role yang mencakup kebijakan IAM. Kebijakan ini harus menentukan izin `GetObject`, `PutObject`, `DeleteObject` dan `ListObject` ke bucket S3 untuk titik akhir target Anda. Peran ini dijalankan baik oleh AWS DMS dan Neptune

dengan akses IAM terhadap kedua target bucket S3 dan basis data Neptune. Untuk informasi lebih lanjut, lihat [Membuat peran layanan IAM untuk mengakses Amazon Neptune sebagai target](#).

Setelah Anda memiliki sumber daya ini, menyiapkan dan memulai migrasi ke target Neptune intinya sama dengan semua migrasi beban penuh menggunakan konsol atau API DMS. Namun, migrasi ke target Neptune memerlukan beberapa langkah unik.

Untuk memigrasi sebuah basis data relasional AWS DMS ke Neptune

1. Buat instans replikasi seperti yang dijelaskan dalam [Membuat instans replikasi](#).
2. Buat dan uji basis data relasional SQL yang didukung oleh AWS DMS untuk titik akhir sumber.
3. Buat dan uji titik akhir target untuk basis data Neptune Anda.

Untuk menghubungkan titik akhir target ke basis data Neptune, tentukan nama server untuk titik akhir klaster Neptune atau titik akhir instans penulis Neptune. Tentukan juga folder bucket S3 untuk AWS DMS yang digunakan untuk menyimpan file perantara untuk beban massal ke basis data Neptune.

Selama migrasi berjalan, AWS DMS menyimpan semua data target yang dimigrasi di dalam folder bucket S3 ini hingga ukuran file maksimum yang Anda tentukan. Ketika penyimpanan file ini mencapai ukuran maksimum tersebut, AWS DMS melakukan pemuatan massal data S3 yang tersimpan ke dalam basis data target. Ia membersihkan folder untuk mengaktifkan penyimpanan data target tambahan apa pun untuk pemuatan berikutnya ke basis data target. Untuk informasi lebih lanjut tentang menentukan pengaturan ini, lihat [Menentukan pengaturan titik akhir untuk Amazon Neptune sebagai target](#).

4. Buat tugas replikasi beban penuh dengan sumber daya yang dibuat dalam langkah 1-3 dan lakukan hal berikut:
 - a. Gunakan pemetaan tabel tugas seperti biasa untuk mengidentifikasi skema sumber tertentu, tabel, dan views untuk bermigrasi dari basis data relasional Anda menggunakan pilihan yang tepat dan aturan transformasi. Untuk informasi lebih lanjut, lihat [Menggunakan pemetaan tabel untuk menentukan pengaturan tugas](#).
 - b. Tentukan pemetaan target dengan memilih salah satu dari berikut ini untuk menentukan aturan pemetaan dari tabel sumber dan tampilan ke grafik basis data target Neptune Anda:

- Gremlin JSON - Untuk informasi tentang penggunaan Gremlin JSON untuk memuat basis data Neptune, lihat [Format memuat data Gremlin](#) di Panduan Pengguna Amazon Neptune.
 - SPARQL RDB to Resource Description Framework Mapping Language (R2RLL) - Untuk informasi tentang menggunakan SPARQL R2RLL, lihat spesifikasi W3C [R2RLL: RDB ke bahasa pemetaan RDF](#).
- c. Lakukan salah satu dari berikut ini:
- Menggunakan konsol AWS DMS, tentukan pilihan pemetaan grafik menggunakan Aturan pemetaan grafik pada halaman Buat tugas migrasi basis data.
 - Menggunakan API AWS DMS, tentukan pilihan ini menggunakan permintaan parameter TaskData dari Panggilan API CreateReplicationTask.

Untuk informasi lebih lanjut dan contoh penggunaan Gremlin JSON dan SPARQL R2RLL dalam menentukan aturan pemetaan grafik, lihat [Menentukan aturan pemetaan grafik menggunakan Gremlin dan R2RLL untuk Amazon Neptune sebagai target](#).

5. Mulai replikasi untuk tugas migrasi Anda.

Menentukan pengaturan titik akhir untuk Amazon Neptune sebagai target

Untuk membuat atau mengubah titik akhir target, Anda dapat menggunakan konsol atau CreateEndpoint atau Operasi API ModifyEndpoint.

Untuk target Neptune di AWS DMS konsol, tentukan Pengaturan khusus Titik Akhir pada halaman konsol Buat titik akhir atau Ubah titik akhir. Untuk CreateEndpoint dan ModifyEndpoint, tentukan parameter permintaan untuk pilihan NeptuneSettings. Contoh berikut menunjukkan cara melakukannya dengan menggunakan CLI.

```
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}'
```

Di sini, pilihan CLI `--server-name` menentukan nama server untuk titik akhir penulis kluster Neptune. Atau Anda dapat menentukan nama server untuk instans titik akhir penulis Neptune.

Berikut adalah parameter permintaan pilihan `--neptune-settings`:

- `ServiceAccessRoleArn` - (Wajib) Amazon Resource Name (ARN) dari peran layanan yang Anda buat untuk titik akhir target Neptune. Untuk informasi lebih lanjut, lihat [Membuat peran layanan IAM untuk mengakses Amazon Neptune sebagai target](#).
- `S3BucketName` - (Wajib) Nama bucket S3 tempat DMS dapat menyimpan sementara data grafik yang dimigrasikan dalam file `.csv` sebelum memuatnya secara massal ke basis data target Neptune. DMS memetakan data sumber SQL data grafik sebelum menyimpannya dalam file `.csv`.
- `S3BucketFolder` - (Wajib) Jalur folder tempat Anda menginginkan DMS untuk menyimpan data grafik yang dimigrasikan di bucket S3 yang ditentukan oleh `S3BucketName`.
- `ErrorRetryDuration` - (Opsional) Jumlah milidetik waktu tunggu DMS untuk mencoba kembali muatan massal data grafik yang dimigrasikan ke basis data target Neptune sebelum dinyatakan error. Default adalah 250.
- `MaxFileSize` - (Opsional) Ukuran maksimum dalam KB dari data grafik yang dimigrasikan lalu disimpan ke dalam file `.csv` sebelum DMS memuat data secara massal ke basis data target Neptune. Default adalah 1.048.576 KB (1 GB). Jika berhasil, DMS akan membersihkan bucket, siap untuk menyimpan batch berikutnya dari data grafik yang dimigrasi.
- `MaxRetryCount` - (Opsional) Frekuensi DMS mencoba kembali muatan massal data grafik yang dimigrasikan ke basis data target Neptune sebelum memunculkan error. Default adalah 5.
- `IAMAuthEnabled` - (Opsional) Jika Anda ingin mengaktifkan otorisasi IAM untuk titik akhir ini, atur parameter ini ke `true` dan lampirkan dokumen kebijakan IAM yang tepat ke peran layanan Anda yang ditentukan oleh `ServiceAccessRoleArn`. Default adalah `false`.

Membuat peran layanan IAM untuk mengakses Amazon Neptune sebagai target

Untuk mengakses Neptune sebagai target, buat peran layanan menggunakan IAM. Tergantung pada konfigurasi titik akhir Neptune Anda, lampirkan peran ini pada beberapa atau semua dokumen kepercayaan dan kebijakan IAM berikut. Ketika Anda membuat titik akhir Neptune, peran layanan

ini diberikan pada ARN. Melakukan hal ini memungkinkan AWS DMS dan Amazon Neptune untuk mengambil izin untuk mengakses Neptune dan bucket Amazon S3 yang terkait.

Jika Anda mengatur parameter `IAMAuthEnabled` dalam `NeptuneSettings` ke `true` di konfigurasi titik akhir Neptune Anda, lampirkan kebijakan IAM seperti berikut ke peran layanan Anda. Jika Anda mengatur `IAMAuthEnabled` ke `false`, Anda dapat mengabaikan kebijakan ini.

```
// 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/*"
    }
  ]
}
```

Kebijakan IAM sebelumnya memungkinkan akses penuh ke klaster target Neptune yang ditentukan oleh `Resource`.

Lampirkan kebijakan IAM seperti berikut ke peran layanan Anda. Kebijakan ini memungkinkan DMS untuk secara sementara menyimpan data grafik yang dimigrasi di bucket S3 yang Anda buat untuk pemuatan massal ke basis data target 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/"
    ]
  }
]
}

```

Kebijakan IAM sebelumnya memungkinkan akun Anda untuk meminta isi bucket S3 (`arn:aws:s3:::my-bucket`) yang dibuat untuk target Neptune Anda. Hal ini juga memungkinkan akun Anda untuk beroperasi sepenuhnya pada isi semua file bucket dan folder (`arn:aws:s3:::my-bucket/`).

Edit hubungan kepercayaan dan lampirkan IAM role berikut ke peran layanan Anda untuk mengizinkan AWS DMS dan layanan basis data Amazon Neptune untuk mengambil peran tersebut.

```

{
  "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"
  }
]
```

Untuk informasi tentang penentuan peran layanan ini pada titik akhir target Neptune, lihat [Menentukan pengaturan titik akhir untuk Amazon Neptune sebagai target](#).

Menentukan aturan pemetaan grafik menggunakan Gremlin dan R2RLL untuk Amazon Neptune sebagai target

Aturan pemetaan grafik yang Anda buat menentukan bagaimana data yang diekstrak dari sumber basis data relasional SQL kemudian dimuat ke target kluster basis data Neptune. Format aturan pemetaan ini berbeda tergantung pada apakah aturan untuk memuat data grafik properti menggunakan ApacheTinkerPopGremlin atau Resource Description Framework (RDF) data menggunakan R2RML. Setelah itu, Anda dapat menemukan informasi tentang format ini dan tempat untuk mempelajari selengkapnya.

Anda dapat menentukan aturan pemetaan ini ketika membuat tugas migrasi menggunakan konsol atau API DMS.

Menggunakan konsol, tentukan aturan pemetaan ini dengan menggunakan Aturan pemetaan grafik pada halaman Buat tugas migrasi basisdata. Di Aturan pemetaan grafik, Anda dapat memasukkan dan mengedit aturan pemetaan secara langsung menggunakan editor yang disediakan. Atau Anda dapat menelusuri file yang berisi aturan pemetaan dalam format pemetaan grafik yang sesuai.

Menggunakan API, tentukan pilihan ini berdasarkan permintaan parameter `TaskData` dari Panggilan API `CreateReplicationTask`. Atur `TaskData` ke jalur file yang berisi aturan pemetaan dalam format pemetaan grafik yang sesuai.

Aturan pemetaan grafik untuk menghasilkan data property-graph menggunakan Gremlin

Menggunakan Gremlin untuk menghasilkan data property-graph, menentukan objek JSON dengan aturan pemetaan untuk setiap entitas grafik yang akan dihasilkan dari data sumber. Format JSON ini ditetapkan secara khusus untuk pemuatan massal Amazon Neptune. Templat berikut menunjukkan nampak seperti apa setiap aturan dalam objek ini.

```

{
  "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)"
        }
    ]
}

```

Kehadiran label vertex menyiratkan bahwa pembuatan vertex sedang terjadi di sini. Ketidadaannya menyiratkan bahwa vertex dibuat oleh sumber yang berbeda, dan definisi ini hanya menambahkan sifat vertex. Tentukan sebanyak mungkin definisi vertex dan edge yang diperlukan untuk menentukan pemetaan untuk seluruh sumber basis data relasional Anda.

Contoh aturan untuk tabel employee berikut.

```

{
  "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"
        }
      ]
    }
  ]
}

```

Di sini, definisi vertex dan edge memetakan hubungan pelaporan dari node employee dengan ID karyawan (EmpID) dan node employee dengan ID manajer (managerId).

Untuk informasi lebih lanjut tentang cara membuat aturan pemetaan grafik menggunakan Gremlin JSON, lihat [Format memuat data Gremlin](#) di Panduan Pengguna Amazon Neptune.

Aturan pemetaan grafik untuk menghasilkan data RDF/SPARQL

Jika Anda memuat data RDF untuk dikueri menggunakan SPARQL, tuliskan aturan pemetaan grafik di R2RLL. R2RLL adalah bahasa W3C standar untuk pemetaan data relasional untuk RDF. Dalam satu file R2RMM, sebuah Peta tiga kali lipat (misalnya, pada <#TriplesMap1> berikut)

menentukan aturan untuk menerjemahkan setiap baris dari tabel logis ke nol atau lebih tiga kali lipat RDF. Sebuah Peta subjek (misalnya, apa pun dari `rr:subjectMap` berikut) menentukan aturan untuk menghasilkan subyek dari tiga kali lipat RDF yang dihasilkan oleh peta tiga kali lipat. Sebuah peta predikat-objek (misalnya, apa pun dari `rr:predicateObjectMap` berikut) adalah fungsi yang menciptakan satu atau lebih pasangan predikat-objek untuk setiap baris tabel logis dari tabel logis.

Contoh sederhana untuk tabel nodes berikut.

```
@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" ];
  ]
```

Pada contoh sebelumnya, pemetaan mendefinisikan node grafik terpetakan dari tabel karyawan.

Contoh sederhana lain untuk tabel Student berikut.

```
@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" ]
```

```
].
```

Pada contoh sebelumnya, pemetaan mendefinisikan pemetaan node grafikfriend-of-a-friendhubungan antara orang-orang dalamStudentmeja.

Untuk informasi lebih lanjut tentang membuat aturan pemetaan grafik menggunakan SPARQL R2RLL, lihat spesifikasi W3C [R2RLL: RDB ke bahasa pemetaan RDF](#).

Tipe data untuk migrasi Gremlin dan R2RLL ke Amazon Neptune sebagai target

AWS DMS melakukan pemetaan tipe data dari titik akhir sumber SQL Anda ke target Neptune Anda menggunakan salah satu dari dua cara. Cara yang digunakan tergantung pada format pemetaan grafik yang Anda gunakan untuk memuat basis data Neptune:

- ApacheTinkerPopGremlin, menggunakan representasi JSON dari data migrasi.
- SPARQL W3C menggunakan representasi R2RLL dari data migrasi.

Untuk informasi lebih lanjut tentang dua format pemetaan grafik ini, lihat [Menentukan aturan pemetaan grafik menggunakan Gremlin dan R2RLL untuk Amazon Neptune sebagai target](#).

Selanjutnya, Anda dapat menemukan deskripsi dari pemetaan tipe data untuk setiap format.

Sumber SQL untuk pemetaan ke tipe data target Gremlin

Tabel berikut menunjukkan pemetaan tipe data dari sumber SQL ke target yang sudah diformat GREMLIN.

AWS DMS memetakan semua tipe data sumber SQL yang tidak terdaftar ke Gremlin String.

Tipe data sumber SQL	Tipe data target Gremlin
NUMERIC (dan varian)	Double
DECIMAL	
TINYINT	Byte
SMALLINT	Short
INT, INTEGER	Int

Tipe data sumber SQL	Tipe data target Gremlin
BIGINT	Long
FLOAT	Float
DOUBLE PRECISION	
REAL	Double
BIT	Boolean
BOOLEAN	
DATE	Date
TIME	
TIMESTAMP	
CHARACTER (dan varian)	String

Untuk informasi lebih lanjut tentang tipe data Gremlin untuk memuat Neptune, lihat [Jenis data Gremlin](#) di Panduan Pengguna Neptune.

Sumber SQL untuk pemetaan ke tipe data target R2RML (RDF)

Tabel berikut menunjukkan pemetaan tipe data dari sumber SQL ke target yang diformat R2RML.

Semua tipe data RDF yang terdaftar adalah case-sensitive, kecuali RDF literal. AWS DMS memetakan tipe data sumber SQL yang tidak terdaftar ke RDF literal.

Sebuah RDF literal adalah salah satu dari berbagai bentuk leksikal literal dan tipe data. Untuk informasi lebih lanjut, lihat [literal RDF](#) dalam spesifikasi W3C Resource Description Framework (RDF): Konsep dan Abstrak.

Tipe data sumber SQL	Tipe data target R2RXML (RDF)
BINARY (dan varian)	xsd:hexBinary

Tipe data sumber SQL	Tipe data target R2RXML (RDF)
NUMERIC (dan varian) DECIMAL	xsd:decimal
TINYINT SMALLINT INT, INTEGER BIGINT	xsd:integer
FLOAT DOUBLE PRECISION REAL	xsd:double
BIT BOOLEAN	xsd:boolean
DATE	xsd:date
TIME	xsd:time
TIMESTAMP	xsd:dateTime
CHARACTER (dan varian)	RDF literal

Untuk informasi lebih lanjut tentang tipe data RDF yang digunakan memuat Neptune dan pemetaan mereka ke tipe data sumber SQL, lihat [Konversi datatype](#) dalam spesifikasi W3C R2RLL: RDB ke Pemetaan Bahasa RDF.

Keterbatasan menggunakan Amazon Neptune sebagai target

Batasan berikut berlaku saat menggunakan Amazon Neptune sebagai target:

- AWS DMS saat ini mendukung tugas beban penuh hanya untuk migrasi ke target Neptune. Migrasi Perubahan pengambilan data (CDC) ke target Neptune tidak didukung.
- Pastikan basis data Neptune target Anda dihapus secara manual dari semua data sebelum memulai tugas migrasi, seperti pada contoh berikut.

Untuk drop semua data (vertex dan edge) dalam grafik, jalankan perintah Gremlin berikut.

```
gremlin> g.V().drop().iterate()
```

Untuk drop vertex yang memiliki label 'customer', jalankan perintah Gremlin berikut.

```
gremlin> g.V().hasLabel('customer').drop()
```

Note

Dibutuhkan beberapa waktu untuk menjatuhkan set data besar. Anda mungkin ingin mengulangi `drop()` dengan batas, misalnya, `limit(1000)`.

Untuk drop edge yang memiliki label 'rated', jalankan perintah Gremlin berikut.

```
gremlin> g.E().hasLabel('rated').drop()
```

Note

Dibutuhkan beberapa waktu untuk menjatuhkan set data besar. Anda mungkin ingin mengulangi `drop()` dengan batas, misalnya `limit(1000)`.

- Operasi API DMS `DescribeTableStatistics` dapat mengembalikan hasil yang tidak akurat tentang tabel yang diberikan karena sifat struktur data grafik Neptune.

Selama migrasi, AWS DMS memindai setiap tabel sumber dan menggunakan pemetaan grafik untuk mengonversi sumber data menjadi grafik Neptune. Data yang dikonversi pertama kali disimpan dalam folder bucket S3 yang ditentukan untuk titik akhir target. Jika sumbernya dipindai dan data S3 perantara ini berhasil dibuat, `DescribeTableStatistics` mengasumsikan bahwa data berhasil dimuat ke dalam basis data target Neptune. Tapi ini tidak selalu benar. Untuk

memverifikasi bahwa data dimuat dengan benar untuk tabel tertentu, bandingkan nilai kembalian `count()` di kedua ujung migrasi untuk tabel tersebut.

Pada contoh berikut, AWS DMS telah memuat sebuah tabel `customer` dari basis data sumber, yang diberi label 'customer' dalam grafik basis data target Neptune. Anda dapat memastikan bahwa label ini ditulis ke basis data target. Untuk melakukannya bandingkan jumlah `customer` baris yang tersedia dari basis data sumber dengan jumlah 'customer' baris berlabel yang dimuat dalam basis data target Neptune setelah tugas selesai.

Untuk mendapatkan jumlah baris pelanggan yang tersedia dari basis data sumber menggunakan SQL, jalankan perintah berikut.

```
select count(*) from customer;
```

Untuk mendapatkan jumlah 'customer' baris berlabel yang dimuat ke dalam grafik basis data target menggunakan Gremlin, jalankan yang berikut ini.

```
gremlin> g.V().hasLabel('customer').count()
```

- Saat ini, jika ada satu tabel tunggal yang gagal memuat, maka seluruh tugas gagal. Tidak seperti dalam target basis data relasional, data di Neptune sangat terhubung, yang dalam banyak kasus membuatnya tidak mungkin untuk melanjutkan tugas. Jika tugas tidak dapat dilanjutkan dengan sukses karena tipe kegagalan memuat data seperti ini, maka buatlah tugas baru untuk memuat tabel yang gagal dimuat. Sebelum menjalankan tugas baru ini, secara manual kosongkan tabel sebagian dimuat dari target Neptune.

Note

Anda dapat melanjutkan tugas yang gagal migrasi ke target Neptune jika kegagalan masih dapat dipulihkan (misalnya, kesalahan transit jaringan).

- AWS DMS mendukung sebagian besar standar untuk R2RLL. Namun AWS DMS tidak mendukung standar R2RLL tertentu, termasuk inverse expression, joins, dan views. Sebuah solusi untuk view R2RLL adalah dengan membuat view SQL kustom yang sesuai dalam basis data sumber. Dalam tugas migrasi, gunakan pemetaan tabel untuk memilih view sebagai masukan. Kemudian petakantampilan ke tabel yang kemudian dikonsumsi oleh R2RLM untuk menghasilkan data grafik.

- Ketika Anda memigrasi data sumber dengan tipe data SQL yang tidak didukung, data target yang dihasilkan dapat kehilangan ketepatan. Untuk informasi selengkapnya, lihat [Tipe data untuk migrasi Gremlin dan R2RLL ke Amazon Neptune sebagai target](#).
- AWS DMS tidak mendukung migrasi data LOB ke target Neptunus.

Menggunakan Redis sebagai target AWS Database Migration Service

Redis adalah penyimpanan struktur data sumber terbuka di memori digunakan sebagai basis data, cache, dan pialang pesan. Mengelola data dalam memori dapat mengakibatkan operasi baca atau tulis membutuhkan waktu kurang dari satu milidetik, dan ratusan juta operasi dilakukan setiap detik. Sebagai penyimpanan data dalam memori, Redis mendukung aplikasi yang paling menuntut yang membutuhkan waktu respons sub-milidetik.

Dengan menggunakan AWS DMS, Anda dapat memigrasi data dari database sumber yang didukung ke penyimpanan data Redis target dengan waktu henti minimal. Untuk informasi tambahan tentang Redis lihat, [Redis Dokumentasi](#).

Selain Redis lokal, AWS Database Migration Service mendukung hal berikut:

- [Amazon ElastiCache for Redis](#) sebagai penyimpanan data target. ElastiCache untuk Redis bekerja dengan klien Redis Anda dan menggunakan format data Redis terbuka untuk menyimpan data Anda.
- [Amazon MemoryDB untuk Redis sebagai penyimpanan data](#) target. MemoryDB kompatibel dengan Redis dan memungkinkan Anda untuk membangun aplikasi menggunakan semua struktur data Redis, API, dan perintah yang digunakan saat ini.

Untuk informasi tambahan tentang bekerja dengan Redis sebagai target untuk AWS DMS, lihat bagian berikut:

Topik

- [Prasyarat untuk menggunakan kluster Redis sebagai target untuk AWS DMS](#)
- [Keterbatasan saat menggunakan Redis sebagai target untuk AWS Database Migration Service](#)
- [Migrasi data dari database relasional atau non-relasional ke target Redis](#)
- [Menentukan pengaturan titik akhir untuk Redis sebagai target](#)

Prasyarat untuk menggunakan klaster Redis sebagai target untuk AWS DMS

DMS mendukung target Redis lokal dalam konfigurasi mandiri, atau sebagai cluster Redis di mana data secara otomatis sharded di beberapa node. Sharding adalah proses memisahkan data menjadi potongan-potongan kecil yang disebut pecahan yang tersebar di beberapa server atau node. Akibatnya, pecahan adalah partisi data yang berisi subset dari total kumpulan data, dan menyajikan sepotong beban kerja keseluruhan.

Karena Redis adalah penyimpanan data NoSQL kunci-nilai, konvensi penamaan kunci Redis untuk digunakan ketika sumber Anda adalah database relasional, adalah `schema-name.table-name.primary-key`. Dalam Redis, kunci dan nilai tidak harus berisi karakter khusus%. Jika tidak, DMS melewati catatan.

Note

Jika Anda menggunakan ElastiCache untuk Redis sebagai target, DMS mendukung mode cluster diaktifkan konfigurasi saja. Untuk informasi selengkapnya tentang penggunaan ElastiCache Redis versi 6.x atau yang lebih tinggi untuk membuat penyimpanan data target yang diaktifkan mode klaster, lihat [Memulai di Panduan Pengguna Amazon ElastiCache for Redis](#).

Sebelum memulai migrasi database, luncurkan klaster Redis Anda dengan kriteria berikut.

- Cluster Anda memiliki satu atau beberapa shard.
- Jika Anda menggunakan target ElastiCache for Redis, pastikan klaster Anda tidak menggunakan kontrol akses berbasis peran IAM. Sebagai gantinya, gunakan Redis Auth untuk mengautentikasi pengguna.
- Aktifkan Multi-AZ (Availability Zone).
- Pastikan cluster memiliki memori yang cukup tersedia untuk menyesuaikan dengan data yang akan dimigrasi dari basis data Anda.
- Pastikan klaster Redis target Anda dihapus dari semua data sebelum memulai tugas migrasi awal.

Anda harus menentukan persyaratan keamanan untuk migrasi data sebelum membuat konfigurasi klaster. DMS mendukung migrasi untuk menargetkan kelompok replikasi terlepas dari konfigurasi enkripsi mereka. Tetapi Anda dapat mengaktifkan atau menonaktifkan enkripsi hanya ketika Anda membuat konfigurasi klaster Anda.

Keterbatasan saat menggunakan Redis sebagai target untuk AWS Database Migration Service

Keterbatasan berikut berlaku saat menggunakan Redis sebagai target:

- Karena Redis adalah key-value no-sql data store, konvensi penamaan kunci Redis untuk digunakan ketika sumber Anda adalah database relasional, adalah `schema-name.table-name.primary-key`
- Dalam Redis, kunci-nilai tidak dapat berisi karakter khusus. % Jika tidak, DMS melewatkan catatan.
- DMS tidak akan memigrasi baris yang berisi karakter khusus.
- DMS tidak akan memigrasi bidang yang berisi karakter khusus dalam nama bidang.
- Mode LOB penuh tidak didukung.
- Otoritas Sertifikat pribadi (CA) tidak didukung saat menggunakan ElastiCache untuk Redis sebagai target.

Migrasi data dari database relasional atau non-relasional ke target Redis

Anda dapat memigrasi data dari sumber SQL atau penyimpanan data NoSQL langsung ke target Redis. Menyiapkan dan memulai migrasi ke target Redis intinya sama dengan semua migrasi pengambilan data menggunakan konsol DMS atau API DMS. Untuk melakukan migrasi database ke target Redis, Anda melakukan hal berikut.

- Membuat instans replikasi untuk melakukan semua proses untuk migrasi. Untuk informasi selengkapnya, lihat [Membuat instans replikasi](#).
- Tentukan titik akhir sumber. Untuk informasi selengkapnya, lihat [Membuat titik akhir sumber dan target](#).
- Temukan nama DNS dan nomor port cluster Anda.
- Unduh paket sertifikat yang dapat Anda gunakan untuk memverifikasi sambungan SSL.
- Tentukan titik akhir target, seperti yang dijelaskan di bawah ini.
- Membuat tugas atau serangkaian tugas untuk menentukan tabel dan proses replikasi apa yang ingin Anda gunakan. Untuk informasi selengkapnya, lihat [Membuat tugas](#).
- Migrasi data dari basis data sumber ke klaster target Anda.

Anda memulai migrasi basis data dengan salah satu dari dua cara berikut:

1. Anda dapat memilih AWS DMS konsol dan melakukan setiap langkah di sana.
2. Anda dapat menggunakan AWS Command Line Interface (AWS CLI). Untuk informasi lebih lanjut tentang menggunakan CLI dengan AWS DMS, lihat [AWS CLI untuk AWS DMS](#).

Untuk menemukan nama DNS dan nomor port klaster Anda

- Gunakan AWS CLI perintah berikut untuk memberikan nama grup replikasi Anda.
`replication-group-id`

```
aws elasticache describe-replication-groups --replication-group-id myreplgroup
```

Di sini, output menunjukkan nama DNS dalam Address atribut dan nomor port dalam Port atribut node utama dalam cluster.

```
...  
"ReadEndpoint": {  
  "Port": 6379,  
  "Address": "myreplgroup-  
111.1abc1d.1111.uuu1.cache.example.com"  
}  
...
```

Jika Anda menggunakan MemoryDB untuk Redis sebagai target Anda, gunakan AWS CLI perintah berikut untuk memberikan alamat endpoint ke klaster Redis Anda.

```
aws memorydb describe-clusters --clusterid clusterid
```

Download bundel sertifikat untuk digunakan untuk memverifikasi koneksi SSL

- Masukkan wget perintah berikut pada baris perintah. Wget adalah GNU baris perintah utilitas alat gratis yang digunakan untuk men-download file dari internet.

```
wget https://s3.aws-api-domain/rds-downloads/rds-combined-ca-bundle.pem
```

Di sini, *aws-api-domain* melengkapi domain Amazon S3 di AWS Wilayah Anda yang diperlukan untuk mengakses bucket S3 khusus dan *rds-combined-ca-bundle* file.pem yang disediakan.

Untuk membuat titik akhir target menggunakan konsol AWS DMS

Endpoint ini adalah untuk target Redis Anda yang sudah berjalan.

- Pada konsol tersebut, memilih Titik Akhir dari panel navigasi dan kemudian memilih Membuat Titik Akhir. Tabel berikut menjelaskan pengaturan.

Untuk opsi ini	Lakukan ini
Jenis titik akhir	Pilih jenis titik akhir Target.
Pengenal titik akhir	Masukkan nama titik akhir Anda. Misalnya, sertakan jenis titik akhir dalam nama, seperti my-redis-target .
Target mesin	Pilih Redis sebagai jenis mesin database yang Anda inginkan endpoint ini untuk terhubung.
Nama klaster	Masukkan nama DNS cluster Redis Anda.
Pelabuhan	Masukkan nomor port klaster Redis Anda.
Protokol keamanan SSL	<p>Pilih salah satu teks biasa atau enkripsi SSL.</p> <p>Teks biasa —Opsi ini tidak menyediakan enkripsi Transport Layer Security (TLS) untuk lalu lintas antara titik akhir dan database.</p> <p>Enkripsi SSL —Jika Anda memilih opsi ini, masukkan sertifikat Otoritas Sertifikat SSL (CA) ARN untuk memverifikasi sertifikat server dan membuat koneksi terenkripsi.</p>

Untuk opsi ini	Lakukan ini
	Untuk Redis lokal, DMS mendukung Otoritas Sertifikat (CA) publik dan swasta. Untuk ElastiCache untuk Redis, DMS hanya mendukung CA publik.
Jenis otentikasi	<p>Pilih jenis otentikasi yang akan dilakukan saat menghubungkan ke Redis. Opsi termasuk, Tidak Ada, Peran otentikasi, dan token Otentikasi.</p> <p>Jika Anda memilih Peran otentikasi, berikan nama pengguna Autentikasi dan kata sandi otentikasi.</p> <p>Jika Anda memilih Token otentikasi, berikan kata sandi otentikasi saja.</p>
Contoh replikasi	[Opsional] Hanya jika Anda berniat untuk menguji koneksi Anda, pilih nama contoh replikasi yang sebelumnya Anda masukkan pada halaman Create replikasi instance.

Setelah Anda selesai memberikan semua informasi untuk titik akhir Anda, AWS DMS membuat titik akhir target Redis Anda untuk digunakan selama migrasi basis data.

Untuk informasi tentang membuat tugas migrasi dan memulai migrasi database, lihat [Membuat tugas](#).

Menentukan pengaturan titik akhir untuk Redis sebagai target

Untuk membuat atau mengubah titik akhir target, Anda dapat menggunakan konsol atau `CreateEndpoint` atau Operasi API `ModifyEndpoint`.

Untuk target Redis di AWS DMS konsol, tentukan pengaturan khusus titik akhir pada halaman Buat titik akhir atau Ubah konsol titik akhir.

Saat menggunakan `CreateEndpoint` dan operasi `ModifyEndpoint` API, tentukan parameter permintaan untuk `RedisSettings` opsi tersebut. Contoh berikut menunjukkan cara melakukannya dengan menggunakan 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"
    }
  }
}
```

--redis-settingsParameternya mengikuti:

- **ServerName**— (Diperlukan) Dari jenisstring, menentukan klaster Redis bahwa data akan dimigrasi ke, dan dalam VPC yang sama Anda.
- **Port**- (Diperlukan) Dari jenisnumber, nilai port yang digunakan untuk mengakses titik akhir.
- **SslSecurityProtocol**- (Opsional) Nilai yang valid termasuk plaintext danssl-encryption. Defaultnya adalah ssl-encryption.

plaintextOpsi ini tidak menyediakan enkripsi Transport Layer Security (TLS) untuk lalu lintas antara endpoint dan database.

Gunakan ssl-encryption untuk membuat koneksi terenkripsi. ssl-encryptiontidak memerlukan SSL Certificate Authority (CA) ARN untuk memverifikasi sertifikat server, tetapi satu

dapat diidentifikasi secara opsional menggunakan pengaturan. `SslCaCertificateArn` Jika ARN otoritas sertifikat tidak diberikan, DMS menggunakan CA root Amazon.

Saat menggunakan target Redis lokal, Anda dapat menggunakannya `SslCaCertificateArn` untuk mengimpor Otoritas Sertifikat (CA) publik atau pribadi ke DMS, dan menyediakan ARN tersebut untuk otentikasi server. CA pribadi tidak didukung saat menggunakan ElastiCache untuk Redis sebagai target.

- `AuthType`- (Diperlukan) Menunjukkan jenis otentikasi yang akan dilakukan saat menghubungkan ke Redis. Nilai yang valid termasuk `none`, `auth-token`, dan `auth-role`.

`auth-token`Opsi ini membutuhkan "`AuthPassword`" disediakan, sedangkan `auth-role` opsi membutuhkan "`AuthUserName`" dan "`AuthPassword`" disediakan.

Menggunakan Babelfish sebagai target untuk AWS Database Migration Service

Anda dapat memigrasi data dari database sumber Microsoft SQL Server ke target Babelfish menggunakan AWS Database Migration Service

Babelfish untuk Aurora PostgreSQL memperluas database Edisi Kompatibel Amazon Aurora PostgreSQL Anda dengan kemampuan untuk menerima koneksi database dari klien Microsoft SQL Server. Melakukan hal ini memungkinkan aplikasi awalnya dibangun untuk SQL Server untuk bekerja secara langsung dengan Aurora PostgreSQL dengan beberapa perubahan kode dibandingkan dengan migrasi tradisional, dan tanpa mengubah driver database.

Untuk informasi tentang versi Babelfish yang AWS DMS mendukung sebagai target, lihat [Target untuk AWS DMS](#) Versi Babelfish sebelumnya di Aurora PostgreSQL memerlukan peningkatan sebelum menggunakan titik akhir Babelfish.

Note

Titik akhir target Aurora PostgreSQL adalah cara yang lebih disukai untuk memigrasi data ke Babelfish. Untuk informasi selengkapnya, lihat [Menggunakan Babelfish untuk Aurora PostgreSQL sebagai target](#).

Untuk informasi tentang penggunaan Babelfish sebagai titik akhir database, lihat [Babelfish for Aurora PostgreSQL di Panduan Pengguna Amazon Aurora untuk Aurora](#)

Prasyarat untuk menggunakan Babelfish sebagai target AWS DMS

Anda harus membuat tabel sebelum memigrasi data untuk memastikan bahwa AWS DMS menggunakan tipe data yang benar dan metadata tabel. Jika Anda tidak membuat tabel pada target sebelum menjalankan migrasi, AWS DMS dapat membuat tabel dengan jenis data yang salah dan izin. Misalnya, AWS DMS membuat kolom timestamp sebagai biner (8) sebagai gantinya, dan tidak menyediakan fungsionalitas timestamp/rowversion yang diharapkan.

Untuk mempersiapkan dan membuat tabel Anda sebelum migrasi

1. Jalankan pernyataan membuat tabel DDL Anda yang mencakup kendala unik, kunci primer, atau kendala default.

Jangan menyertakan kendala kunci asing, atau pernyataan DDL untuk objek seperti tampilan, prosedur yang disimpan, fungsi, atau pemicu. Anda dapat menerapkannya setelah memigrasi database sumber Anda.

2. Mengidentifikasi setiap kolom identitas, kolom dihitung, atau kolom yang berisi rowversion atau timestamp jenis data untuk tabel Anda. Kemudian, buat aturan transformasi yang diperlukan untuk menangani masalah yang diketahui saat menjalankan tugas migrasi. Untuk informasi lebih lanjut lihat, [Aturan dan tindakan transformasi](#).
3. Identifikasi kolom dengan tipe data yang tidak didukung Babelfish. Kemudian, ubah kolom yang terpengaruh di tabel target untuk menggunakan tipe data yang didukung, atau buat aturan transformasi yang menghapusnya selama tugas migrasi. Untuk informasi lebih lanjut lihat, [Aturan dan tindakan transformasi](#).

Tabel berikut mencantumkan tipe data sumber yang tidak didukung oleh Babelfish, dan jenis data target yang direkomendasikan sesuai untuk digunakan.

Jenis data sumber	Tipe data Babelfish yang direkomendasikan
PUSAKA	NVARCHAR (250)
GEOMETRY	VARCHAR (MAKS)
GEOGRAPHY	VARCHAR (MAKS)

Untuk mengatur level unit kapasitas Aurora (ACU) untuk basis data sumber V2 Tanpa Server Aurora PostgreSQL Anda

Anda dapat meningkatkan kinerja tugas AWS DMS migrasi sebelum menjalankannya dengan menetapkan nilai ACU minimum.

- Dari jendela pengaturan kapasitas Severless v2, tetapkan ACU Minimum ke 2, atau level yang wajar untuk klaster Aurora DB Anda.

Untuk informasi tambahan tentang pengaturan unit kapasitas Aurora, lihat [Memilih rentang kapasitas v2 Tanpa Server Aurora untuk klaster Aurora di Panduan Pengguna Amazon Aurora](#)

Setelah menjalankan tugas AWS DMS migrasi, Anda dapat mengatur ulang nilai minimum ACU Anda ke tingkat yang wajar untuk basis data sumber V2 Tanpa Server Aurora PostgreSQL Anda.

Persyaratan keamanan saat menggunakan Babelfish sebagai target AWS Database Migration Service

Berikut ini menjelaskan persyaratan keamanan untuk digunakan AWS DMS dengan target Babelfish:

- Nama pengguna administrator (pengguna Admin) yang digunakan untuk membuat database.
- Login PSQL dan pengguna dengan izin SELECT, INSERT, UPDATE, DELETE, dan REFERENCES yang cukup.

Izin pengguna untuk menggunakan Babelfish sebagai target untuk AWS DMS

Important

Untuk tujuan keamanan, akun pengguna yang digunakan untuk migrasi data harus merupakan pengguna terdaftar dalam database Babelfish apa pun yang Anda gunakan sebagai target.

Titik akhir target Babelfish Anda memerlukan izin pengguna minimum untuk menjalankan migrasi.
AWS DMS

Untuk membuat login dan pengguna Transact-SQL (T-SQL) dengan hak istimewa rendah

1. Buat login dan kata sandi untuk digunakan saat menghubungkan ke server.

```
CREATE LOGIN dms_user WITH PASSWORD = 'password';  
GO
```

2. Buat database virtual untuk klaster Babelfish Anda.

```
CREATE DATABASE my_database;  
GO
```

3. Buat pengguna T-SQL untuk database target Anda.

```
USE my_database  
GO  
CREATE USER dms_user FOR LOGIN dms_user;  
GO
```

4. Untuk setiap tabel dalam database Babelfish Anda, izin GRANT ke tabel.

```
GRANT SELECT, DELETE, INSERT, REFERENCES, UPDATE ON [dbo].[Categories] TO dms_user;
```

Batasan dalam menggunakan Babelfish sebagai target untuk AWS Database Migration Service

Batasan berikut berlaku saat menggunakan database Babelfish sebagai target untuk: AWS DMS

- Hanya mode persiapan meja “Do Nothing “yang didukung.
- Tipe data ROWVERSION memerlukan aturan pemetaan tabel yang menghapus nama kolom dari tabel selama tugas migrasi.
- Tipe data sql_variant tidak didukung.
- Mode LOB penuh didukung. Menggunakan SQL Server sebagai titik akhir sumber memerlukan SQL Server Endpoint Connection Atribut pengaturan ForceFullLob=True yang akan ditetapkan agar LOB untuk dimigrasi ke titik akhir target.
- Pengaturan tugas replikasi memiliki batasan berikut:

```
{  
  "FullLoadSettings": {  
    "TargetTablePrepMode": "DO_NOTHING",
```

```

    "CreatePkAfterFullLoad": false,
  }.
}

```

- TIME (7), DATETIME2 (7), dan DATETIMEOFFSET (7) tipe data di Babelfish membatasi nilai presisi untuk bagian detik waktu untuk 6 digit. Pertimbangkan untuk menggunakan nilai presisi 6 untuk tabel target Anda saat menggunakan tipe data ini. Untuk Babelfish versi 2.2.0 dan yang lebih tinggi, saat menggunakan TIME (7) dan DATETIME2 (7), digit ketujuh presisi selalu nol.
- Dalam mode DO_NOTHING, DMS memeriksa untuk melihat apakah tabel sudah ada. Jika tabel tidak ada dalam skema target, DMS membuat tabel berdasarkan definisi tabel sumber, dan memetakan setiap jenis data yang ditetapkan pengguna ke tipe data dasar mereka.
- Tugas AWS DMS migrasi ke target Babelfish tidak mendukung tabel yang memiliki kolom menggunakan tipe data ROWVERSION atau TIMESTAMP. Anda dapat menggunakan aturan pemetaan tabel yang menghapus nama kolom dari tabel selama proses transfer. Dalam contoh aturan transformasi berikut, tabel bernama Actor dalam sumber Anda diubah untuk menghapus semua kolom dimulai dengan karakter col dari Actor tabel di target Anda.

```

{
  "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%"
    }
  }
]}

```

}

- Untuk tabel dengan identitas atau kolom yang dihitung, di mana tabel target menggunakan nama kasus campuran seperti Kategori, Anda harus membuat tindakan aturan transformasi yang mengubah nama tabel menjadi huruf kecil untuk tugas DMS Anda. Contoh berikut menunjukkan cara membuat tindakan aturan transformasi, Membuat huruf kecil menggunakan konsol. AWS DMS Untuk informasi selengkapnya, lihat [Aturan dan tindakan transformasi](#).

Transformation rules

You can use transformation rules to change or transform schema, table or column names of some or all of the selected objects. [Info](#)

[Add transformation rule](#)

▼ where **schema name** is like 'dbo' and **table name** is like '%', convert-lowercase 📄 ✕

Rule target

Table ▼

Source name

Enter a schema ▼

Source name
Use the % character as a wildcard

dbo

Table name
Use the % character as a wildcard

%

Action

Make lowercase ▼

- Sebelum Babelfish versi 2.2.0, DMS membatasi jumlah kolom yang dapat Anda replikasi ke titik akhir target Babelfish menjadi dua puluh (20) kolom. Dengan Babelfish 2.2.0 batasnya meningkat menjadi 100 kolom. Tetapi dengan Babelfish versi 2.4.0 dan lebih tinggi, jumlah kolom yang dapat Anda replikasi meningkat lagi. Anda dapat menjalankan contoh kode berikut terhadap database SQL Server Anda untuk menentukan tabel yang terlalu panjang.

```
USE myDB;
GO
```

```

DECLARE @Babelfish_version_string_limit INT = 8000; -- Use 380 for Babelfish versions
before 2.2.0
WITH bfindpoint
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 bfindpoint
WHERE IsTooLong = 1
ORDER BY TABLE_SCHEMA, InsertIntoCommandLength DESC, TABLE_NAME
;

```

Tipe data target untuk Babelfish

Tabel berikut menunjukkan tipe data target Babelfish yang didukung saat menggunakan AWS DMS dan pemetaan default dari AWS DMS tipe data.

Untuk informasi tambahan tentang tipe data AWS DMS, lihat [Tipe data untuk AWS Database Migration Service](#).

Tipe data AWS DMS	Tipe data babelfish
BOOLEAN	TINYINT
BYTES	VARBINARY(length)
TANGGAL	DATE
TIME	WAKTU
INT1	SMALLINT
INT2	SMALLINT
INT4	INT
INT8	BIGINT
NUMERIK	NUMERIK (p, s)
REAL4	REAL
REAL8	FLOAT
STRING	<p>Jika kolom berupa kolom tanggal atau waktu, lakukan hal berikut:</p> <ul style="list-style-type: none"> • Untuk SQL Server 2008 dan yang lebih tinggi, gunakan DATETIME2. • Untuk versi sebelumnya, jika skalanya 3 atau kurang, gunakan DATETIME. Selain itu gunakan VARCHAR (37). <p>Jika kolom bukan berupa kolom tanggal atau waktu, gunakan VARCHAR (length).</p>
UINT1	TINYINT
UINT2	SMALLINT

Tipe data AWS DMS	Tipe data babelfish
UINT4	INT
UINT8	BIGINT
WSTRING	NVARCHAR (panjang)
BLOB	VARBINARY(max) Untuk menggunakan tipe data ini dengan DMS, Anda harus mengaktifkan penggunaan gumpalan untuk tugas tertentu. DMS mendukung tipe data BLOB hanya dalam tabel yang menyertakan kunci primer.
CLOB	VARCHAR(max) Untuk menggunakan tipe data ini dengan DMS, Anda harus mengaktifkan penggunaan CLOB untuk tugas tertentu.
NCLOB	NVARCHAR(max) Untuk menggunakan tipe data ini dengan DMS, Anda harus mengaktifkan penggunaan NCLOB untuk tugas tertentu. Selama CDC, DMS mendukung tipe data NCLOB hanya dalam tabel yang menyertakan kunci primer.

Menggunakan Amazon Timestream sebagai target AWS Database Migration Service

Anda dapat menggunakan AWS Database Migration Service untuk memigrasikan data dari database sumber ke titik akhir target Amazon Timestream, dengan dukungan untuk migrasi data Beban Penuh dan CDC.

Amazon Timestream adalah layanan database deret waktu yang cepat, terukur, dan tanpa server yang dibuat untuk konsumsi data volume tinggi. Data deret waktu adalah urutan titik data yang

dikumpulkan selama interval waktu, dan digunakan untuk mengukur peristiwa yang berubah seiring waktu. Ini digunakan untuk mengumpulkan, menyimpan, dan menganalisis metrik dari aplikasi IoT DevOps, aplikasi, dan aplikasi analitik. Setelah Anda memiliki data di Timestream, Anda dapat memvisualisasikan dan mengidentifikasi tren dan pola dalam data Anda dalam waktu dekat. Untuk informasi tentang Amazon Timestream, lihat [Apa itu Amazon Timestream?](#) di Panduan Pengembang Amazon Timestream.

Topik

- [Prasyarat untuk menggunakan Amazon Timestream sebagai target AWS Database Migration Service](#)
- [Pengaturan tugas beban penuh multithread](#)
- [Pengaturan beban tugas CDC multithread](#)
- [Pengaturan titik akhir saat menggunakan Timestream sebagai target AWS DMS](#)
- [Membuat dan memodifikasi titik akhir target Amazon Timestream](#)
- [Menggunakan pemetaan objek untuk memigrasikan data ke topik Timestream](#)
- [Batasan saat menggunakan Amazon Timestream sebagai target AWS Database Migration Service](#)

Prasyarat untuk menggunakan Amazon Timestream sebagai target AWS Database Migration Service

Sebelum Anda mengatur Amazon Timestream sebagai target AWS DMS, pastikan Anda membuat peran IAM. Peran ini harus memungkinkan AWS DMS untuk mendapatkan akses ke data yang sedang dimigrasikan ke Amazon Timestream. Set minimum izin akses untuk peran yang Anda gunakan untuk bermigrasi ke Timestream ditampilkan dalam kebijakan IAM berikut.

```
{
  "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"
}
]
}

```

Jika Anda bermaksud untuk memigrasikan semua tabel, gunakan * untuk *TABLE_NAME* pada *contoh* di atas.

Perhatikan hal berikut tentang penggunaan Timestream sebagai target:

- Jika Anda bermaksud untuk menelan data historis dengan stempel waktu melebihi 1 tahun, sebaiknya gunakan untuk menulis data AWS DMS ke Amazon S3 dalam format nilai dipisahkan koma (csv). Kemudian, gunakan pemuatan batch Timestream untuk menyerap data ke Timestream. Untuk informasi selengkapnya, lihat [Menggunakan pemuatan batch di Timestream](#) dalam panduan pengembang [Amazon Timestream](#).
- Untuk migrasi data beban penuh data yang berusia kurang dari 1 tahun, sebaiknya setelah periode penyimpanan penyimpanan memori tabel Timestream lebih besar dari atau sama dengan stempel waktu tertua. Kemudian, setelah migrasi selesai, edit penyimpanan penyimpanan memori tabel ke nilai yang diinginkan. Misalnya, untuk memigrasikan data dengan stempel waktu tertua yang berumur 2 bulan, lakukan hal berikut:
 - Setelah retensi penyimpanan memori tabel target Timestream menjadi 2 bulan.

- Mulai migrasi data menggunakan AWS DMS.
- Setelah migrasi data selesai, ubah periode retensi tabel Timestream target ke nilai yang Anda inginkan.

Kami merekomendasikan untuk memperkirakan biaya penyimpanan memori sebelum migrasi, menggunakan informasi pada halaman berikut:

- [Harga Amazon Timestream](#)
- [AWSkalkulator harga](#)
- Untuk migrasi data CDC, sebaiknya atur periode retensi penyimpanan memori dari tabel target sehingga data yang tertelan berada dalam batas retensi penyimpanan memori. Untuk informasi selengkapnya, lihat [Menulis Praktik Terbaik](#) di panduan [pengembang Amazon Timestream](#).

Pengaturan tugas beban penuh multithread

Untuk membantu meningkatkan kecepatan transfer data, AWS DMS mendukung tugas migrasi beban penuh multithreaded ke titik akhir target Timestream dengan pengaturan tugas ini:

- `MaxFullLoadSubTasks` - Gunakan pilihan ini untuk menunjukkan jumlah maksimum tabel sumber yang dimuat secara paralel. DMS memuat setiap tabel ke dalam tabel target Amazon Timestream yang sesuai menggunakan subtugas khusus. Default adalah 8; nilai maksimum adalah 49.
- `ParallelLoadThreads`— Gunakan opsi ini untuk menentukan jumlah utas yang AWS DMS digunakan untuk memuat setiap tabel ke dalam tabel target Amazon Timestream. Nilai maksimum untuk target Timestream adalah 32. Anda dapat meminta untuk meningkatkan batas maksimum ini.
- `ParallelLoadBufferSize`— Gunakan opsi ini untuk menentukan jumlah maksimum catatan yang akan disimpan dalam buffer yang digunakan thread beban paralel untuk memuat data ke target Amazon Timestream. Nilai default adalah 50. Nilai maksimumnya adalah 1.000. Gunakan pengaturan ini dengan `ParallelLoadThreads`. `ParallelLoadBufferSize` hanya berlaku bila ada lebih dari satu thread.
- `ParallelLoadQueuesPerThread` - Gunakan pilihan ini untuk menentukan jumlah antrian yang diakses setiap thread secara bersamaan untuk mengambil catatan data dari antrian dan menghasilkan beban batch untuk target. Default adalah 1. Namun, untuk target Amazon Timestream dengan berbagai ukuran muatan, rentang yang valid adalah 5—512 antrian per utas.

Pengaturan beban tugas CDC multithread

Untuk mempromosikan performa CDC, AWS DMS mendukung pengaturan tugas ini:

- `ParallelApplyThreads`— Menentukan jumlah thread bersamaan yang AWS DMS digunakan selama beban CDC untuk mendorong catatan data ke titik akhir target Timestream. Nilai defaultnya adalah 0 dan nilai maksimumnya adalah 32.
- `ParallelApplyBufferSize`— Menentukan jumlah maksimum catatan untuk disimpan di setiap antrian buffer untuk thread bersamaan untuk mendorong ke titik akhir target Timestream selama beban CDC. Nilai default adalah 100 dan nilai maksimum adalah 1.000. Gunakan pilihan ini saat `ParallelApplyThreads` menentukan lebih dari satu thread.
- `ParallelApplyQueuesPerThread`— Menentukan jumlah antrian yang diakses setiap thread untuk mengeluarkan catatan data dari antrian dan menghasilkan beban batch untuk titik akhir Timestream selama CDC. Nilai default adalah 1 dan nilai maksimum adalah 512.

Pengaturan titik akhir saat menggunakan Timestream sebagai target AWS DMS

Anda dapat menggunakan pengaturan titik akhir untuk mengonfigurasi database target Timestream Anda mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir target menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--timestream-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan endpoint yang dapat Anda gunakan dengan Timestream sebagai target.

Nama	Penjelasan
<code>MemoryDuration</code>	<p>Setel atribut ini untuk menentukan retensi terikat untuk menyimpan data yang dimigrasikan di penyimpanan memori Timestream. Waktu diukur dalam satuan jam. Penyimpanan memori Timestream dioptimalkan untuk throughput konsumsi tinggi dan akses cepat.</p> <p>Nilai default: 24 (jam)</p> <p>Nilai valid: 1 hingga 8.736 (1 jam hingga 12 bulan diukur dalam jam)</p>

Nama	Penjelasan
	Contoh: <code>--timestream-settings '{"MemoryDuration": 20}'</code>
DatabaseName	Tetapkan atribut ini untuk menentukan nama database Timestream target. Tipe: string Contoh: <code>--timestream-settings '{"DatabaseName": " db_name"}</code>
TableName	Tetapkan atribut ini untuk menentukan nama tabel Timestream target. Tipe: string Contoh: <code>--timestream-settings '{"TableName": " table_name "}'</code>
MagneticDuration	Atur atribut ini untuk menentukan durasi magnetik yang diterapkan ke tabel Timestream dalam beberapa hari. Ini adalah retensi yang terikat untuk data yang dicerna. Timestream menghapus stempel waktu yang melebihi batas retensi. Untuk informasi selengkapnya, lihat Penyimpanan di Panduan Pengembang Amazon Timestream . Contoh: <code>--timestream-settings '{"MagneticDuration": "3"}</code>

Nama	Penjelasan
CdcInsertsAndUpdates	<p>Setel atribut ini <code>true</code> untuk menentukan bahwa AWS DMS hanya menerapkan sisipan dan pembaruan, dan bukan menghapus. Timestream tidak mengizinkan penghapusan catatan, jadi jika nilai ini <code>false</code>, AWS DMS membatalkan catatan yang sesuai dalam database Timestream daripada menghapusnya. Untuk informasi lebih lanjut, lihat Batasan berikut ini.</p> <p>Nilai default: <code>false</code></p> <p>Contoh: <code>--timestream-settings '{"CdcInsertsAndUpdates": "true"}'</code></p>
EnableMagneticStoreWrites	<p>Setel atribut ini <code>true</code> untuk mengaktifkan penulisan penyimpanan magnetik. Ketika nilai ini <code>false</code>, AWS DMS tidak menulis catatan dengan stempel waktu yang lebih tua dari periode penyimpanan memori dari tabel target, karena Timestream tidak mengizinkan penulisan penyimpanan magnetik secara default. Untuk informasi selengkapnya, lihat Menulis Praktik Terbaik di Panduan Pengembang Amazon Timestream.</p> <p>Nilai default: <code>false</code></p> <p>Contoh: <code>--timestream-settings '{"EnableMagneticStoreWrites": "true"}'</code></p>

Membuat dan memodifikasi titik akhir target Amazon Timestream

Setelah Anda membuat peran IAM dan menetapkan set minimum izin akses, Anda dapat membuat titik akhir target Amazon Timestream menggunakan konsol, atau dengan menggunakan AWS DMS perintah di, dengan `create-endpoint` sintaks [AWS CLI/JSON](#). `--timestream-settings` `'{"EndpointSetting": "value", ...}'`

Contoh berikut menunjukkan cara membuat dan memodifikasi titik akhir target Timestream menggunakan. AWS CLI

Buat perintah titik akhir target Timestream

```
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,
}
```

Ubah perintah titik akhir target Timestream

```
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,
}
```

Menggunakan pemetaan objek untuk memigrasikan data ke topik Timestream

AWS DMS menggunakan aturan pemetaan tabel untuk memetakan data dari sumber ke topik Timestream target. Untuk memetakan data ke topik target, Anda menggunakan jenis aturan pemetaan tabel yang disebut pemetaan objek. Anda menggunakan pemetaan objek untuk menentukan bagaimana catatan data dalam peta sumber ke catatan data yang dipublikasikan ke topik Timestream.

Topik Timestream tidak memiliki struktur preset selain memiliki kunci partisi.

Note

Anda tidak perlu menggunakan pemetaan objek. Anda dapat menggunakan pemetaan tabel biasa untuk berbagai transformasi. Namun, jenis kunci partisi akan mengikuti perilaku default ini:

- Primary Key digunakan sebagai kunci partisi untuk Full Load.
- Jika tidak ada pengaturan tugas penerapan paralel yang digunakan, `schema.table` digunakan sebagai kunci partisi untuk CDC.
- Jika pengaturan tugas penerapan paralel digunakan, kunci utama digunakan sebagai kunci partisi untuk CDC.

Untuk membuat aturan pemetaan objek, tetapkan `rule-type` sebagai `object-mapping`. Aturan ini menentukan jenis pemetaan objek yang ingin Anda gunakan. Struktur untuk aturan tersebut adalah sebagai berikut.

```
{
  "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 saat ini mendukung `map-record-to-record` dan `map-record-to-document` sebagai satu-satunya nilai valid untuk parameter `rule-action`. Nilai `map-record-to-record` dan `map-record-to-document` menentukan apa yang dilakukan AWS DMS secara default pada catatan yang tidak dikecualikan sebagai bagian dari daftar atribut `exclude-columns`. Nilai-nilai ini tidak mempengaruhi pemetaan atribut dengan cara apapun.

Gunakan `map-record-to-record` saat bermigrasi dari database relasional ke topik Timestream. Jenis aturan ini menggunakan `taskResourceId.schemaName.tableName` nilai dari database relasional sebagai kunci partisi dalam topik Timestream dan menciptakan atribut untuk setiap kolom dalam database sumber. Saat menggunakan `map-record-to-record`, untuk kolom apa pun dalam tabel sumber yang tidak tercantum dalam daftar atribut `exclude-columns`, AWS DMS menciptakan atribut yang sesuai dalam topik target. Atribut yang sesuai ini dibuat terlepas dari apakah kolom sumber tersebut digunakan dalam pemetaan atribut.

Salah satu cara untuk memahami `map-record-to-record` adalah melihatnya beraksi. Untuk contoh ini, anggaplah bahwa Anda memulai dengan baris tabel basis data relasional dengan struktur dan data berikut.

FirstName	LastName	StoreId	HomeAddress	HomePhone	WorkAddress	WorkPhone	DateofBirth
Randy	Marsh	5	221B Baker Street	1234567890	31 Spooner	9876543210	02/29/1988

FirstName	LastName	StoreId	HomeAddress	HomePhone	WorkAddress	WorkPhone	DateofBirth
					Street, Quahog		

Untuk memigrasikan informasi ini dari skema bernama Test ke topik Timestream, Anda membuat aturan untuk memetakan data ke topik target. Aturan berikut menggambarkan pemetaan.

```
{
  "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"
      }
    }
  ]
}
```

Diberikan topik Timestream dan kunci partisi (dalam hal ini, `taskResourceId.schemaName.tableName`), berikut ini menggambarkan format rekaman yang dihasilkan menggunakan data sampel kami dalam topik target Timestream:

```
{
  "FirstName": "Randy",
```

```
"LastName": "Marsh",
"StoreId": "5",
"HomeAddress": "221B Baker Street",
"HomePhone": "1234567890",
"WorkAddress": "31 Spooner Street, Quahog",
"WorkPhone": "9876543210",
"DateOfBirth": "02/29/1988"
}
```

Batasan saat menggunakan Amazon Timestream sebagai target AWS Database Migration Service

Batasan berikut berlaku saat menggunakan Amazon Timestream sebagai target:

- **Dimensi dan Stempel Waktu:** Timestream menggunakan dimensi dan stempel waktu dalam data sumber seperti kunci primer komposit, dan juga tidak memungkinkan Anda untuk meningkatkan nilai-nilai ini. Ini berarti bahwa jika Anda mengubah stempel waktu atau dimensi untuk catatan dalam database sumber, database Timestream akan mencoba membuat catatan baru. Dengan demikian mungkin bahwa jika Anda mengubah dimensi atau stempel waktu rekaman sedemikian rupa sehingga cocok dengan catatan lain yang ada, maka AWS DMS memperbarui nilai catatan lain alih-alih membuat catatan baru atau memperbarui catatan terkait sebelumnya.
- **Perintah DDL:** Rilis saat ini AWS DMS hanya mendukung CREATE TABLE dan perintah DROP TABLE DDL.
- **Batasan Rekam:** Timestream memiliki batasan untuk catatan seperti ukuran rekaman dan ukuran ukuran. Untuk informasi selengkapnya, lihat [Kuota](#) di [Panduan Developer Amazon Timestream](#).
- **Menghapus Catatan dan Nilai Null:** Timestream tidak mendukung penghapusan catatan. Untuk mendukung migrasi catatan yang dihapus dari sumber, AWS DMS menghapus bidang terkait dalam catatan dalam database target Timestream. AWS DMS mengubah nilai di bidang catatan target yang sesuai dengan 0 untuk bidang numerik, null untuk bidang teks, dan false untuk bidang boolean.
- Timestream sebagai target tidak mendukung sumber yang bukan database relasional (RDBMS).
- AWS DMS hanya mendukung Timestream sebagai target di wilayah berikut:
 - AS Timur (N. Virginia)
 - AS Timur (Ohio)
 - AS Barat (Oregon)
 - Europe (Ireland)

- Europe (Frankfurt)
- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- Timestream sebagai target tidak mendukung pengaturan `TargetTablePrepMode` ke `TRUNCATE_BEFORE_LOAD`. Kami merekomendasikan penggunaan `DROP_AND_CREATE` untuk pengaturan ini.

Menggunakan Amazon RDS untuk Db2 dan IBM Db2 LUW sebagai target AWS DMS

Anda dapat memigrasikan data ke Amazon RDS untuk Db2 atau database Db2 lokal dari database Db2 LUW menggunakan (). AWS Database Migration Service AWS DMS

Untuk informasi tentang versi Db2 LUW yang AWS DMS mendukung sebagai target, lihat. [Target untuk AWS DMS](#)

Anda dapat menggunakan Lapisan Soket Aman (SSL) untuk mengenkripsi sambungan antara titik akhir Db2 LUW dan instans replikasi. Untuk informasi selengkapnya tentang penggunaan SSL dengan titik akhir Db2 LUW, lihat. [Menggunakan SSL dengan AWS Database Migration Service](#)

Keterbatasan saat menggunakan Db2 LUW sebagai target untuk AWS DMS

Keterbatasan berikut berlaku saat menggunakan database Db2 LUW sebagai target untuk. AWS DMS Untuk batasan penggunaan Db2 LUW sebagai sumber, lihat. [Keterbatasan saat menggunakan Db2 LUW sebagai sumber untuk AWS DMS](#)

- AWS DMS hanya mendukung Db2 LUW sebagai target ketika sumbernya adalah Db2 LUW atau Db2 untuk z/OS.
- Menggunakan Db2 LUW sebagai target tidak mendukung replikasi dengan mode LOB penuh.
- Menggunakan Db2 LUW sebagai target tidak mendukung tipe data XML dalam fase beban penuh. Ini adalah batasan utilitas IBM dbload. Untuk informasi selengkapnya, lihat [Utilitas dbload dalam dokumentasi](#) IBM Informix Servers.
- AWS DMS memotong bidang BLOB dengan nilai yang sesuai dengan karakter kutipan ganda ("). Ini adalah batasan utilitas IBM dbload.

Pengaturan titik akhir saat menggunakan Db2 LUW sebagai target untuk AWS DMS

Anda dapat menggunakan pengaturan titik akhir untuk mengonfigurasi basis data target Db2 LUW Anda mirip dengan menggunakan atribut koneksi tambahan. Anda menentukan pengaturan saat Anda membuat titik akhir target menggunakan AWS DMS konsol, atau dengan menggunakan `create-endpoint` perintah di [AWS CLI](#), dengan sintaks `--ibm-db2-settings '{"EndpointSetting": "value", ...}'` JSON.

Tabel berikut menunjukkan pengaturan endpoint yang dapat Anda gunakan dengan Db2 LUW sebagai target.

Nama	Penjelasan
KeepCsvFiles	Jika benar, AWS DMS menyimpan file.csv apa pun ke target Db2 LUW yang digunakan untuk mereplikasi data. DMS menggunakan file-file ini untuk analisis dan pemecahan masalah..
LoadTimeout	Jumlah waktu (dalam milidetik) sebelum AWS DMS waktu habis operasi yang dilakukan oleh DMS pada target Db2. Nilai default adalah 1200 (20 menit).
MaxFileSize	Menentukan ukuran maksimum (dalam KB) dari file.csv yang digunakan untuk mentransfer data ke Db2 LUW.
WriteBufferSize	Ukuran (dalam KB) buffer tulis file dalam memori yang digunakan saat membuat file.csv pada disk lokal pada instance replikasi DMS. Nilai default adalah 1024 (1 MB).

Mengkonfigurasi titik akhir VPC sebagaiAWSSumber DMS dan titik akhir target

AWSDMS mendukung titik akhir Amazon Virtual Private Cloud (VPC) sebagai sumber dan target.AWS DMS dapat terhubung keAWSsumber atau basis data target dengan titik akhir Amazon VPC selama rute yang ditentukan secara eksplisit ke sumber dan basis data target ini ditentukan di dalamnyaAWSDMS VPC.

Dengan mendukung titik akhir Amazon VPC,AWSDMS membuatnya lebih mudah untuk mempertahankan end-to-end keamanan jaringan untuk semua tugas replikasi tanpa konfigurasi dan pengaturan jaringan tambahan. Menggunakan titik akhir VPC untuk semua titik akhir sumber dan target memastikan bahwa semua lalu lintas Anda tetap berada dalam VPC Anda dan di bawah kendali Anda. Upgrade keAWSDMS versi 3.4.7 dan yang lebih tinggi mengharuskan Anda mengonfigurasiAWSDMS untuk menggunakan titik akhir VPC atau menggunakan rute publik ke semua sumber dan titik akhir target Anda yang berinteraksi dengan Amazon Web Services berikut:

- Amazon S3
- Amazon Kinesis
- AWS Secrets Manager
- Amazon DynamoDB
- Amazon Redshift
- Amazon OpenSearch Layanan

Anda mungkin memerlukan titik akhir VPC untuk mendukungAWSDMS dimulai dengan versi 3.4.7 seperti yang dijelaskan berikut.

Siapa yang terpengaruh saat bermigrasi keAWSDMS versi 3.4.7 dan lebih tinggi?

Anda terkena dampak jika Anda menggunakan satu atau lebih dari yang sebelumnya terdaftarAWSTitik akhir DMS, dan titik akhir ini tidak dapat dirutekan secara publik atau mereka tidak memiliki titik akhir VPC yang sudah terkait dengannya.

Siapa yang tidak terpengaruh saat bermigrasi keAWSDMS versi 3.4.7 dan lebih tinggi?

Anda tidak terpengaruh jika:

- Anda tidak menggunakan satu atau lebih dari yang terdaftar sebelumnyaAWSTitik akhir DMS.
- Anda menggunakan salah satu titik akhir yang terdaftar sebelumnya dan dapat dirutekan secara publik.
- Anda menggunakan salah satu titik akhir yang terdaftar sebelumnya dan mereka memiliki titik akhir VPC yang terkait dengannya.

Mempersiapkan migrasi keAWS DMS versi 3.4.7 dan lebih tinggi

Untuk mencegahAWS Kegagalan tugas DMS saat Anda menggunakan salah satu titik akhir yang dijelaskan sebelumnya, ambil salah satu langkah berikut sebelum meningkatkanAWS DMS ke versi 3.4.7 atau lebih tinggi:

- Buat yang terkena dampakAWS Titik akhir DMS dapat dirutekan secara publik. Misalnya, tambahkan rute Internet Gateway (IGW) ke VPC apa pun yang sudah digunakan oleh AndaAWS Instance replikasi DMS untuk membuat semua titik akhir sumber dan targetnya dapat dirutekan secara publik.
- Buat titik akhir VPC untuk mengakses semua titik akhir sumber dan target yang digunakan olehAWS DMS seperti yang dijelaskan berikut.

Untuk titik akhir VPC yang ada yang Anda gunakan untukAWS Sumber DMS dan titik akhir target, memastikan bahwa mereka menggunakan kebijakan kepercayaan yang sesuai dengan dokumen kebijakan XML `dms-vpc-role`. Untuk informasi lebih lanjut tentang dokumen kebijakan XML ini, lihat [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#).

Jika tidak, konfigurasi instance replikasi Anda sebagai titik akhir VPC dengan menambahkan titik akhir VPC ke VPC yang mengandungnya. Jika Anda mengonfigurasi instans replikasi tanpa titik akhir publik, menambahkan titik akhir VPC yang dapat diakses publik ke VPC yang berisi instance replikasi Anda membuatnya dapat diakses secara publik. Anda tidak perlu melakukan apa pun lagi untuk mengaitkan instance replikasi Anda dengan titik akhir VPC.

Note

Layanan yang berbeda mungkin memiliki konfigurasi titik akhir VPC yang unik. Misalnya, saat menggunakanAWS Secrets Manager, Anda biasanya tidak perlu menyesuaikan tabel routing. Selalu periksa persyaratan khusus untuk setiap layanan.

Buat titik akhir VPC pada VPC yang berisi instance replikasi Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon VPC di <https://console.aws.amazon.com/vpc/>.
2. Di bilah menu konsol VPC, pilih yang samaWilayah AWS sebagai AndaAWS Contoh replikasi DMS.

3. Di panel navigasi VPC, pilih Titik akhir.
4. Di Titik akhir, pilih Buat titik akhir.
5. Anda dapat menentukan tag nama. Sebagai contoh, **my-endpoint-DynamoDB-01**.
6. Di bawah Jasa untuk S3 atau DynamoDB saja, pilih Nama Layanan dengan Tipe disetel ke Pintu gerbang.
7. Di bawah VPC, pilih VPC yang sama dengan kami AWS Contoh replikasi DMS untuk membuat titik akhir.
8. Di bawah Tabel Rute, pilih semua yang tersedia Tabel Rute ID nilai.
9. Untuk menentukan kontrol akses, di bawah Kebijakan, pilih Akses penuh. Jika Anda ingin menggunakan alat pembuatan kebijakan untuk menentukan kontrol akses Anda sendiri, pilih Kustom. Dalam hal apapun, gunakan kebijakan kepercayaan yang sesuai dengan dokumen kebijakan JSON, `dms-vpc-role`. Untuk informasi lebih lanjut tentang dokumen kebijakan ini, lihat [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#).
10. Di bawah Titik akhir, verifikasi bahwa titik akhir VPC Anda yang baru dibuat Status adalah Tersedia.

Untuk informasi lebih lanjut tentang mengkonfigurasi titik akhir VPC untuk AWS Contoh replikasi DMS, lihat [Konfigurasi jaringan untuk migrasi basis data](#). Untuk informasi lebih lanjut tentang membuat titik akhir VPC antarmuka untuk mengakses AWS Layanan umumnya, lihat [Mengakses sebuah AWS layanan menggunakan antarmuka VPC endpoint](#) di AWS PrivateLink Panduan. Untuk informasi di AWS Ketersediaan regional DMS untuk titik akhir VPC, lihat [AWS Tabel Wilayah](#).

Pernyataan DDL didukung oleh AWS DMS

Anda dapat menjalankan pernyataan bahasa definisi data (DDL) pada basis data sumber selama proses migrasi data. Semua pernyataan ini dilakukan replikasi ke basis data target oleh server replikasi.

Pernyataan DDL yang didukung meliputi berikut ini:

- MEMBUAT TABEL
- Membatalkan tabel
- Mengubah Nama Tabel
- Memotong tabel
- Menambahkan kolom

- Membatalkan kolom
- Mengubah Nama Kolom
- Mengubah jenis data kolom

DMS tidak menangkap semua pernyataan DDL yang didukung untuk beberapa tipe mesin sumber. Dan DMS menangani pernyataan DDL secara berbeda ketika menerapkannya ke mesin target tertentu. Untuk informasi yang mendukung pernyataan DDL untuk sumber tertentu, dan bagaimana pernyataan tersebut diterapkan pada suatu target, lihat topik dokumentasi khusus untuk titik akhir sumber dan target.

Anda dapat menggunakan pengaturan tugas untuk mengkonfigurasi cara DMS menangani perilaku DDL selama proses change data capture (CDC). Untuk informasi selengkapnya, lihat [Pengaturan tugas untuk perubahan pengolahan DDL penanganan](#).

Bekerja dengan AWS DMS tugas

Tugas AWS Database Migration Service (AWS DMS) adalah tempat semua pekerjaan dilakukan. Anda menentukan tabel (atau tampilan), skema, dan pemrosesan khusus yang digunakan untuk migrasi Anda, seperti persyaratan logging, data tabel kontrol, dan penanganan kesalahan.

Sebuah tugas dapat terdiri dari tiga tahap utama:

- Migrasi data yang ada (Full load)
- Aplikasi perubahan cache
- Replikasi yang sedang berlangsung (Ubah Pengambilan Data)

Untuk informasi selengkapnya dan ikhtisar tentang cara tugas AWS DMS migrasi memigrasikan data, lihat [Tampilan tingkat tinggi AWS DMS](#)

Saat membuat tugas migrasi, ada beberapa hal yang perlu Anda ketahui:

- Sebelum membuat tugas, pastikan bahwa Anda membuat titik akhir sumber, titik akhir target, dan instans replikasi.
- Anda dapat menentukan banyak pengaturan tugas untuk menyesuaikan tugas migrasi Anda. Anda dapat mengatur ini dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS DMS API. Pengaturan ini mencakup penentuan bagaimana kesalahan migrasi ditangani, pencatatan kesalahan, dan informasi tabel kontrol. Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).
- Segera jalankan setelah Anda membuat tugas. Tabel target dengan definisi metadata yang diperlukan secara otomatis dibuat dan dimuat, sehingga Anda dapat menentukan replikasi yang sedang berjalan.
- Secara default, AWS DMS memulai tugas Anda segera setelah Anda membuatnya. Namun, dalam kejadian tertentu, Anda mungkin ingin menunda memulai tugas. Misalnya, saat menggunakan AWS CLI, Anda mungkin memiliki proses tertentu yang menciptakan tugas serta proses berbeda yang memulai tugas berdasarkan beberapa peristiwa pemicu. Apabila dibutuhkan, Anda bisa menunda awal tugas Anda.
- Anda dapat memantau, menghentikan, atau memulai ulang tugas menggunakan konsol, AWS CLI, atau AWS DMS API. Untuk informasi tentang menghentikan tugas menggunakan AWS DMS API, lihat [StopReplicationTask](#) di [Referensi AWS DMS API](#).

Berikut ini adalah tindakan yang dapat Anda lakukan ketika bekerja dengan AWS DMS tugas.

Tugas	Dokumentasi terkait
<p>Membuat tugas</p> <p>Saat membuat tugas, tentukan sumber, target, dan instans replikasi sekaligus pengaturan migrasi.</p>	<p>Membuat tugas</p>
<p>Membuat tugas replikasi yang sedang berlangsung</p> <p>Anda dapat mengatur tugas untuk melakukan replikasi terus menerus antara sumber dan target.</p>	<p>Membuat tugas untuk replikasi yang sedang berlangsung menggunakan AWS DMS</p>
<p>Menerapkan pengaturan tugas</p> <p>Setiap tugas memiliki pengaturan yang dapat Anda konfigurasi sesuai dengan kebutuhan migrasi basis data Anda. Buat pengaturan ini dalam file JSON atau, dengan pengaturan tertentu, buat pengaturan menggunakan konsol AWS DMS. Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat Contoh pengaturan tugas.</p>	<p>Menentukan setelan tugas untuk tugas AWS Database Migration Service</p>
<p>Menggunakan pemetaan tabel</p> <p>Pemetaan tabel menentukan pengaturan tugas tambahan untuk tabel lewat jenis-jenis aturan tertentu. Aturan-aturan ini memungkinkan Anda menentukan sumber data, skema sumber, tabel dan pandangan,</p>	<p>Aturan seleksi</p> <p>Aturan seleksi dan tindakan</p> <p>Aturan transformasi</p> <p>Aturan dan tindakan transformasi</p> <p>Aturan pengaturan tabel</p>

Tugas	Dokumentasi terkait
<p>data, setiap perubahan tabel dan data yang terjadi selama tugas berjalan, serta pengaturan migrasi tabel dan kolom dari sumber ke target.</p>	<p>Tabel dan koleksi pengaturan aturan dan operasi</p>
<p>Menjalankan penilaian tugas premi</p> <p>Anda dapat mengaktifkan dan menjalankan penilaian tugas pramigrasi yang menampilkan masalah dengan sumber yang didukung serta basis data target yang berpotensi menyebabkan masalah selama proses migrasi. Ini bisa mencakup masalah seperti jenis data yang tidak didukung, indeks tidak cocok, kunci utama, serta pengaturan tugas bertentangan lainnya. Penilaian pramigrasi ini dilakukan sebelum Anda menjalankan tugas untuk mengenali potensi masalah sebelum muncul selama migrasi.</p>	<p>Mengaktifkan dan bekerja dengan penilaian perdana untuk tugas</p>
<p>Validasi data</p> <p>Validasi data adalah pengaturan tugas yang dapat Anda gunakan agar AWS DMS membandingkan data di database target Anda dengan data dari penyimpanan data sumber Anda.</p>	<p>Validasi data AWS DMS.</p>
<p>Memodifikasi tugas</p> <p>Ketika tugas dihentikan, Anda dapat mengubah pengaturan untuk tugas.</p>	<p>Mengubah tugas</p>

Tugas	Dokumentasi terkait
<p>Memindahkan tugas</p> <p>Ketika tugas dihentikan, Anda dapat memindahkan tugas ke instans replikasi yang berbeda.</p>	<p>Memindahkan tugas</p>
<p>Memuat ulang tabel selama tugas</p> <p>Anda dapat memuat ulang tabel selama mengerjakan tugas apabila terjadi kesalahan.</p>	<p>Memuat ulang tabel selama mengerjakan tugas</p>
<p>Menerapkan filter</p> <p>Anda dapat menggunakan filter sumber untuk membatasi jumlah dan jenis catatan yang ditransfer dari sumber ke target. Misalnya, Anda dapat menentukan bahwa hanya karyawan di markas yang dipindahkan ke basis data target. Terapkan filter pada kolom data.</p>	<p>Menggunakan filter sumber</p>
<p>Memantau tugas</p> <p>Ada beberapa cara untuk mendapatkan informasi tentang performa tugas dan tabel yang digunakan oleh tugas.</p>	<p>Pemantauan Tugas AWS DMS</p>
<p>Mengelola log tugas</p> <p>Anda dapat melihat dan menghapus log tugas menggunakan AWS DMS API atau AWS CLI.</p>	<p>Melihat dan mengelola AWS Mencatat tugas DMS</p>

Topik

- [Membuat tugas](#)
- [Membuat tugas untuk replikasi yang sedang berlangsung menggunakan AWS DMS](#)
- [Mengubah tugas](#)
- [Memindahkan tugas](#)
- [Memuat ulang tabel selama mengerjakan tugas](#)
- [Menggunakan pemetaan tabel untuk menentukan pengaturan tugas](#)
- [Menggunakan filter sumber](#)
- [Mengaktifkan dan bekerja dengan penilaian perdana untuk tugas](#)
- [Menentukan data tambahan untuk pengaturan tugas](#)

Membuat tugas

Untuk membuat tugas AWS DMS migrasi, Anda melakukan hal berikut:

- Buat titik akhir sumber, titik akhir target, dan instans replikasi sebelum Anda membuat tugas migrasi.
- Pilih metode migrasi:
 - Memigrasikan data ke basis data target – Proses ini menciptakan file atau tabel dalam basis data target dan secara otomatis mendefinisikan metadata yang diperlukan pada target. Langkah ini juga mengisi tabel dengan data dari sumber. Data dari tabel dimuat secara paralel untuk meningkatkan efisiensi. Proses ini adalah opsi Migrasi data yang ada di AWS Management Console dan dipanggil `Full Load` di API.
 - Menangkap perubahan selama migrasi – Proses ini menangkap perubahan ke basis data sumber yang terjadi saat data sedang bermigrasi dari sumber ke target. Ketika migrasi data yang diminta telah selesai, proses perubahan data capture (CDC) kemudian menerapkan perubahan yang direkam ke basis data target. Perubahan direkam dan diterapkan sebagai unit transaksi tindakan tunggal, dan Anda dapat memperbarui beberapa tabel target yang berbeda sebagai tindakan satu sumber. Pendekatan ini menjamin integritas transaksional dalam basis data target. Proses ini adalah opsi Migrasikan data yang ada dan tiru perubahan yang sedang berlangsung di konsol dan disebut `full-load-and-cdc` dalam API.
 - Mereplikasi hanya perubahan data dalam basis data sumber –Proses ini membaca pemulihan berkas log dalam sistem manajemen basis data sumber (DBMS) dan mengelompokkan bersama-sama entri untuk setiap transaksi. Dalam beberapa kasus, tidak AWS DMS dapat menerapkan perubahan pada target dalam waktu yang wajar (misalnya, jika target tidak dapat

diakses). Dalam kasus ini, AWS DMS buffer perubahan pada server replikasi selama diperlukan. Fitur ini tidak membaca ulang log sumber DBMS, yang dapat memakan banyak waktu. Proses ini adalah opsi Replikasi perubahan data saja di konsol AWS DMS .

- Tentukan bagaimana tugas harus menangani objek biner besar (LOB) dalam sumber. Untuk informasi selengkapnya, lihat [Mengatur dukungan LOB untuk database sumber dalam suatu tugas AWS DMS](#).
- Tentukan pengaturan tugas migrasi. Hal ini termasuk menyiapkan pencatatan, menentukan data apa yang ditulis ke tabel kontrol migrasi, bagaimana kesalahan ditangani, dan pengaturan lainnya. Untuk informasi selengkapnya tentang pengaturan tugas, lihat [Menentukan setelan tugas untuk tugas AWS Database Migration Service](#).
- Mengatur pemetaan tabel untuk menentukan aturan untuk memilih dan menyaring data yang sedang Anda migrasikan. Untuk informasi lebih lanjut tentang pemetaan tabel, lihat [Menggunakan pemetaan tabel untuk menentukan pengaturan tugas](#). Sebelum Anda menentukan pemetaan, pastikan bahwa Anda meninjau bagian dokumentasi dalam pemetaan jenis data untuk basis data sumber dan target Anda.
- Aktifkan dan jalankan penilaian tugas pramigrasi sebelum menjalankan tugas. Untuk informasi selengkapnya tentang penilaian perdana, lihat [Mengaktifkan dan bekerja dengan penilaian perdana untuk tugas](#).
- Tentukan data tambahan yang diperlukan untuk tugas memigrasikan data Anda. Untuk informasi selengkapnya, lihat [Menentukan data tambahan untuk pengaturan tugas](#).

Anda dapat memilih untuk memulai tugas segera setelah Anda selesai menentukan informasi untuk tugas tersebut pada halaman Buat tugas. Atau, Anda dapat memulai tugas dari halaman Dashboard nanti juga.

Prosedur berikut mengasumsikan bahwa Anda telah menentukan instans informasi replikasi serta titik akhir. Untuk informasi selengkapnya tentang pengaturan titik akhir, lihat [Membuat titik akhir sumber dan target](#).

Untuk membuat tugas migrasi

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.

Jika Anda masuk sebagai pengguna AWS Identity and Access Management (IAM), pastikan Anda memiliki izin yang sesuai untuk mengakses AWS DMS. Untuk informasi lebih lanjut tentang izin yang diperlukan, lihat [Izin IAM diperlukan untuk menggunakan AWS DMS](#).

2. Pada panel navigasi, pilih Tugas migrasi database, lalu pilih Buat tugas.
3. Pada halaman tugas Buat migrasi database, di bagian Konfigurasi tugas, tentukan opsi tugas. Tabel berikut menjelaskan pengaturan.

Create database migration task

Task configuration

Task identifier

Type a unique identifier for the task

Descriptive Amazon Resource Name (ARN) - *optional*

A friendly name to override the default DMS ARN. You cannot modify it after creation.

Friendly-ARN-name

Replication instance

Choose a replication instance

Source database endpoint

Choose a source database endpoint

Target database endpoint

Choose a target database endpoint

Migration type [Info](#)

Migrate existing data

Untuk opsi ini	Lakukan hal berikut
Pengidentifikasi tugas	Masukkan Nama tugas.
Nama Sumber Daya Amazon Deskriptif (ARN) - opsional	Nama ramah untuk mengganti AWS DMS ARN default. Anda tidak dapat mengubah nama ini setelah membuat tugas.
Contoh replikasi	Menunjukkan instans replikasi yang akan digunakan.
Titik akhir basis data sumber	Menunjukkan titik akhir sumber yang akan digunakan.
Titik akhir basis data target	Menunjukkan titik akhir target yang akan digunakan.
Jenis migrasi	Pilih metode migrasi yang ingin Anda gunakan. Anda dapat memilih untuk memigrasikan hanya data yang ada ke basis data target atau mengirimkan perubahan yang sedang berlangsung ke basis data target di samping data yang bermigrasi.

4. Di bagian Pengaturan Tugas, tentukan nilai untuk mengedit tugas Anda, mode persiapan tabel target, menghentikan tugas, pengaturan LOB, validasi, dan pencatatan.

Untuk opsi ini	Lakukan hal berikut
Mode pengeditan	Pilih apakah akan menggunakan Wizard atau editor JSON untuk menentukan pengaturan tugas Anda. Jika Anda memilih Wizard, opsi berikut akan ditampilkan.
Mode mulai CDC untuk transaksi sumber	<p>Pengaturan ini hanya terlihat jika Anda memilih Replikasi perubahan data hanya untuk jenis Migrasi di bagian sebelumnya.</p> <p>Nonaktifkan mode mulai CDC kustom — Jika Anda memilih opsi ini, Anda dapat memulai tugas Anda secara otomatis dengan menggunakan opsi Automatic on create berikut, atau secara manual dengan menggunakan konsol.</p>

Untuk opsi ini	Lakukan hal berikut
	Aktifkan mode mulai CDC kustom - Jika Anda memilih opsi ini, Anda dapat menentukan waktu mulai UTC khusus untuk mulai memproses perubahan.

Untuk opsi ini	Lakukan hal berikut
Mode persiapan tabel target	<p>Setelan ini hanya terlihat jika Anda memilih Migrasi data yang ada atau Migrasi data yang ada dan mereplikasi perubahan yang sedang berlangsung untuk jenis Migrasi di bagian sebelumnya.</p> <p>Tidak melakukan apa-apa — Dalam mode Do nothing, AWS DMS mengasumsikan bahwa tabel target telah dibuat sebelumnya pada target. Jika tabel tidak kosong, konflik mungkin terjadi selama migrasi data dan dapat mengakibatkan kesalahan tugas DMS. Jika tabel target tidak ada, DMS yang akan membuat tabel untuk Anda. Struktur tabel Anda tetap seperti apa adanya dan data yang ada akan tetap ada dalam tabel. Mode Tidak melakukan apapun sesuai untuk tugas yang ada hanya ada di CDC ketika tabel target telah ditimbulk dari sumber dan replikasi yang sedang berlangsung diterapkan untuk menjaga sinkronisasi antara sumber dan target. Untuk pra-membuat tabel, Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT). Untuk informasi selengkapnya, lihat Menginstal AWS SCT.</p> <p>Turunkan tabel pada target – Dalam mode Turunkan tabel pada target, AWS DMS menurunkan tabel target dan membuatnya ulang sebelum memulai migrasi. Pendekatan ini memastikan bahwa tabel target kosong saat migrasi dimulai. AWS DMS hanya membuat objek yang diperlukan untuk memigrasikan data secara efisien: tabel, kunci utama, dan dalam beberapa kasus, indeks unik. AWS DMS tidak membuat indeks sekunder, kendala kunci nonprimer, atau default data kolom. Jika Anda melakukan beban penuh ditambah CDC atau tugas di CDC saja, kami sarankan Anda menghentikan sementara proses migrasi. Setelah itu,</p>

Untuk opsi ini	Lakukan hal berikut
	<p>buat indeks sekunder yang mendukung penyaringan untuk update dan hapus pernyataan.</p> <p>Anda mungkin perlu melakukan beberapa konfigurasi terhadap basis data target ketika Anda menggunakan mode Turunkan tabel pada target. Misalnya, untuk target Oracle, tidak AWS DMS dapat membuat skema (pengguna database) untuk alasan keamanan. Dalam hal ini, Anda membuat skema pengguna sehingga AWS DMS dapat membuat tabel saat migrasi dimulai. Untuk sebagian besar jenis target lainnya, AWS DMS buat skema dan semua tabel terkait dengan parameter konfigurasi yang tepat.</p> <p>Memangkas — Dalam mode Pemotongan, AWS DMS potong semua tabel target sebelum migrasi dimulai. Jika tabel target tidak ada, DMS yang akan membuat tabel untuk Anda. Struktur tabel Anda tetap seperti apa adanya namun tabel terpotong pada target. Mode Memotong sesuai untuk beban penuh atau beban penuh ditambah migrasi CDC di mana skema target telah dibuat sebelumnya sebelum migrasi dimulai. Untuk membuat tabel terlebih dahulu, Anda dapat menggunakan AWS SCT. Untuk informasi selengkapnya, lihat Menginstal AWS SCT.</p> <div data-bbox="730 1402 1507 1810" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p> Note</p><p>Jika target Anda adalah MongoDB, mode Truncate tidak memotong tabel pada target. Sebaliknya, mode ini menurunkan koleksi dan menghilangkan semua indeks. Hindari mode Pemotongan saat target Anda adalah MongoDB.</p></div>

Untuk opsi ini	Lakukan hal berikut
<p>Hentikan tugas setelah beban penuh selesai</p>	<p>Pengaturan ini hanya terlihat jika Anda memilih Migrasi data yang ada dan mereplikasi perubahan yang sedang berlangsung untuk jenis Migrasi di bagian sebelumnya.</p> <p>Jangan berhenti – Jangan hentikan tugas tetapi segera terapkan perubahan cache dan lanjutkan pekerjaan.</p> <p>Berhenti sebelum menerapkan perubahan cache - Hentikan tugas sebelum penerapan perubahan cache. Dengan pendekatan ini, Anda dapat menambahkan indeks sekunder yang bisa mempercepat penerapan perubahan.</p> <p>Berhenti setelah menerapkan perubahan cache - Hentikan tugas setelah perubahan cache diterapkan. Dengan menggunakan pendekatan ini, Anda dapat menambahkan kunci asing jika Anda menggunakan aplikasi transaksional.</p>
<p>Sertakan kolom LOB dalam replikasi</p>	<p>Jangan sertakan kolom LOB – Kolom LOB dikecualikan dari migrasi.</p> <p>Mode LOB penuh - Migrasikan LOB lengkap terlepas dari ukurannya. AWS DMS memigrasikan LOB sedikit demi sedikit dalam potongan yang dikontrol oleh parameter ukuran LOB Chunk. Mode ini lebih lambat dibandingkan dengan menggunakan mode LOB Terbatas.</p> <p>Mode LOB terbatas – Memotong LOB ke nilai parameter Ukuran LOB maksimal. Mode ini lebih cepat daripada menggunakan mode LOB penuh.</p>

Untuk opsi ini	Lakukan hal berikut
Ukuran LOB maksimum (kb)	Dalam Mode LOB terbatas, kolom LOB yang melebihi pengaturan Ukuran LOB maksimal dipotong ke nilai Ukuran LOB maksimal yang sudah ditentukan.
Aktifkan validasi	Memungkinkan validasi data, untuk memverifikasi bahwa data dimigrasi secara akurat dari sumber ke target. Untuk informasi selengkapnya, lihat Validasi data AWS DMS .
Aktifkan CloudWatch log	Mengaktifkan pencatatan oleh Amazon CloudWatch.

- Di bagian Penilaian Premigrasi, pilih apakah akan menjalankan penilaian premi. Penilaian premi memperingatkan Anda tentang potensi masalah migrasi sebelum memulai tugas migrasi database Anda. Untuk informasi selengkapnya, lihat [Mengaktifkan dan bekerja dengan penilaian perdana](#).
- Di bagian konfigurasi startup tugas Migrasi, tentukan apakah akan memulai tugas secara otomatis setelah pembuatan.
- Di bagian Tag, tentukan tag apa pun yang Anda butuhkan untuk mengatur tugas Anda. Anda dapat menggunakan tag untuk mengelola peran dan kebijakan IAM Anda, dan melacak biaya DMS Anda. Untuk informasi selengkapnya, lihat [Penandaan sumber daya](#).
- Setelah selesai dengan pengaturan tugas, pilih Buat tugas.

Menentukan setelan tugas untuk tugas AWS Database Migration Service

Setiap tugas memiliki pengaturan yang dapat Anda konfigurasi sesuai dengan kebutuhan migrasi basis data Anda. Anda membuat pengaturan ini dalam file JSON atau, dengan beberapa pengaturan, Anda dapat menentukan pengaturan menggunakan AWS DMS konsol. Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

Ada beberapa jenis utama pengaturan tugas, seperti berikut ini.

Topik

- [Contoh pengaturan tugas](#)
- [Menargetkan pengaturan tugas metadata](#)
- [Pengaturan tugas beban penuh](#)

- [Pengaturan tugas Perjalanan Waktu](#)
- [Pengaturan tugas pengelogan](#)
- [Mengontrol pengaturan tugas tabel](#)
- [Pengaturan tugas buffer pengaliran](#)
- [Mengubah pengaturan penyetelan pemrosesan](#)
- [Pengaturan tugas validasi data](#)
- [Pengaturan tugas untuk perubahan pengolahan DDL penanganan](#)
- [Pengaturan tugas substitusi karakter](#)
- [Pengaturan tugas citra sebelumnya](#)
- [Penanganan kesalahan pengaturan tugas](#)
- [Menyimpan pengaturan tugas](#)

Pengaturan tugas	Dokumentasi terkait
<p>Membuat laporan penilaian tugas</p> <p>Anda dapat membuat laporan penilaian tugas yang menampilkan jenis data yang tidak didukung sehingga dapat menyebabkan masalah selama migrasi. Anda dapat menjalankan laporan ini pada tugas Anda sebelum menjalankan tugas untuk mengetahui potensi masalah.</p>	<p>Mengaktifkan dan bekerja dengan penilaian perdana untuk tugas</p>
<p>Membuat tugas</p> <p>Saat membuat tugas, tentukan sumber, target, dan instans replikasi sekaligus pengaturan migrasi.</p>	<p>Membuat tugas</p>
<p>Membuat tugas replikasi yang sedang berlangsung</p>	<p>Membuat tugas untuk replikasi yang sedang berlangsung menggunakan AWS DMS</p>

Pengaturan tugas	Dokumentasi terkait
<p>Anda dapat mengatur tugas untuk melakukan replikasi terus menerus antara sumber dan target.</p>	
<p>Menerapkan pengaturan tugas</p> <p>Setiap tugas memiliki pengaturan yang dapat Anda konfigurasi sesuai dengan kebutuhan migrasi basis data Anda. Anda membuat pengaturan ini dalam file JSON atau, dengan beberapa pengaturan, Anda dapat menentukan pengaturan menggunakan AWS DMS konsol.</p>	<p>Menentukan setelan tugas untuk tugas AWS Database Migration Service</p>
<p>Validasi data</p> <p>Gunakan validasi data untuk AWS DMS membandingkan data pada penyimpanan data target Anda dengan data dari penyimpanan data sumber Anda.</p>	<p>Validasi data AWS DMS</p>
<p>Memodifikasi tugas</p> <p>Ketika tugas dihentikan, Anda dapat mengubah pengaturan untuk tugas.</p>	<p>Mengubah tugas</p>
<p>Memuat ulang tabel selama tugas</p> <p>Anda dapat memuat ulang tabel selama mengerjakan tugas apabila terjadi kesalahan.</p>	<p>Memuat ulang tabel selama mengerjakan tugas</p>

Pengaturan tugas	Dokumentasi terkait
<p>Menggunakan pemetaan tabel</p> <p>Tabel pemetaan menggunakan beberapa jenis aturan untuk menentukan pengaturan tugas untuk sumber data, skema sumber, data, dan setiap transformasi yang harus terjadi selama tugas.</p>	<p>Aturan Seleksi Aturan seleksi dan tindakan</p> <p>Aturan Transformasi Aturan dan tindakan transformasi</p>
<p>Menerapkan filter</p> <p>Anda dapat menggunakan filter sumber untuk membatasi jumlah dan jenis catatan yang ditransfer dari sumber ke target. Misalnya, Anda dapat menentukan bahwa hanya karyawan di markas yang dipindahkan ke basis data target. Terapkan filter pada kolom data.</p>	<p>Menggunakan filter sumber</p>
<p>Memantau tugas</p> <p>Ada beberapa cara untuk mendapatkan informasi tentang performa tugas dan tabel yang digunakan oleh tugas.</p>	<p>Pemantauan Tugas AWS DMS</p>
<p>Mengelola log tugas</p> <p>Anda dapat melihat dan menghapus log tugas menggunakan AWS DMS API atau AWS CLI.</p>	<p>Melihat dan mengelola AWS Mencatat tugas DMS</p>

Contoh pengaturan tugas

Anda dapat menggunakan salah satu AWS Management Console atau AWS CLI untuk membuat tugas replikasi. Jika Anda menggunakan AWS CLI, Anda mengatur pengaturan

tugas dengan membuat file JSON, lalu menentukan file://URI dari file JSON sebagai [ReplicationTaskSettings](#) parameter operasi. [CreateReplicationTask](#)

Contoh berikut menunjukkan cara menggunakan AWS CLI untuk memanggil `CreateReplicationTask` operasi:

```
aws dms create-replication-task \  
--replication-task-identifier MyTask \  
--source-endpoint-arn arn:aws:dms:us-  
west-2:123456789012:endpoint:ABCDEFGHIJKLMN  
OPQRSTUVWXYZ1234567890ABC \  
--target-endpoint-arn arn:aws:dms:us-  
west-2:123456789012:endpoint:ABCDEFGHIJKLMN  
OPQRSTUVWXYZ1234567890ABC \  
--replication-instance-arn arn:aws:dms:us-  
west-2:123456789012:rep:ABCDEFGHIJKLMN  
OPQRSTUVWXYZ1234567890ABC \  
--migration-type cdc \  
--table-mappings file://tablemappings.json \  
--replication-task-settings file://settings.json
```

Contoh sebelumnya menggunakan file pemetaan tabel yang disebut `tablemappings.json`. Untuk contoh pemetaan tabel, lihat [Menggunakan pemetaan tabel untuk menentukan pengaturan tugas](#).

Pengaturan tugas file JSON dapat terlihat seperti berikut ini.

```
{  
  "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
  }
}

```

Menargetkan pengaturan tugas metadata

Pengaturan metadata target termasuk yang berikut ini. Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

- **TargetSchema** – Nama skema tabel target. Jika opsi metadata ini kosong, skema dari tabel sumber digunakan. AWS DMS secara otomatis menambahkan prefiks pemilik untuk basis data target terhadap semua tabel jika tidak ada penentuan akan skema sumber. Opsi ini harus dibiarkan

kosong untuk titik akhir target bertipe MySQL. Mengganti nama skema dalam pemetaan data lebih diutamakan daripada pengaturan ini.

- Pengaturan LOB - Pengaturan yang menentukan bagaimana objek besar (LOB) dikelola. Jika Anda mengatur `SupportLobs=true`, Anda harus menetapkan salah satu dari berikut ini di `true`:
 - `FullLobMode` – Jika Anda menetapkan opsi ini di `true`, maka Anda harus memasukkan nilai untuk opsi `LobChunkSize`. Masukkan ukuran bagian-bagian LOB dalam kilobyte untuk digunakan ketika mereplikasi data ke target. Kinerja opsi `FullLobMode` bekerja paling baik untuk ukuran LOB yang sangat besar tetapi cenderung menyebabkan pemuatan yang lebih lambat. Nilai yang disarankan untuk `LobChunkSize` adalah 64 kilobyte. Meningkatkan nilai di `LobChunkSize` atas 64 kilobyte dapat menyebabkan kegagalan tugas.
 - `InlineLobMaxSize`— Nilai ini menentukan LOB mana yang AWS DMS ditransfer secara inline selama beban penuh. Mentransfer LOB kecil lebih efisien daripada mencarinya dari tabel sumber. Selama beban penuh, AWS DMS periksa semua LOB dan lakukan transfer inline untuk LOB yang lebih kecil dari `InlineLobMaxSize` AWS DMS mentransfer semua LOB yang lebih besar dari `InlineLobMaxSize` in `FullLobMode`. Nilai default untuk `InlineLobMaxSize` adalah 0 dan kisarannya adalah 1 —102400 kilobyte (100 MB). Tetapkan nilai untuk `InlineLobMaxSize` hanya jika Anda tahu bahwa sebagian besar LOB lebih kecil dari nilai yang ditentukan dalam `InlineLobMaxSize`.
 - `LimitedSizeLobMode` – Jika Anda menetapkan opsi ini di `true`, maka Anda harus memasukkan nilai untuk opsi `LobMaxSize`. Masukkan ukuran maksimum dalam kilobyte untuk LOB satuan. Nilai maksimum yang disarankan untuk `LobMaxSize` adalah 102400 kilobyte (100 MB).

Untuk informasi selengkapnya tentang kriteria menggunakan pengaturan LOB tugas ini, lihat [Mengatur dukungan LOB untuk database sumber dalam suatu tugas AWS DMS](#). Anda juga dapat mengontrol pengelolaan LOB untuk tabel satuan. Untuk informasi selengkapnya, lihat [Tabel dan koleksi pengaturan aturan dan operasi](#).

- `LoadMaxFileSize` – Opsi untuk titik akhir target berbasis CSV seperti MySQL, PostgreSQL, dan Amazon Redshift yang mendukung penggunaan file comma separated value (.csv) untuk memuat data. `LoadMaxFileSize` mendefinisikan ukuran maksimum pada disk data yang disimpan dan dibongkar, seperti file.csv. Opsi ini menimpa atribut koneksi titik akhir target, `maxFileSize`. Anda dapat memberikan nilai dari 0, yang menunjukkan bahwa opsi ini tidak menimpa atribut koneksi, hingga 100.000 KB.
- `BatchApplyEnabled` – Menentukan apakah setiap transaksi diterapkan secara satuan atau perubahan dilakukan dalam batch. Nilai default-nya adalah `false`.

Ketika `BatchApplyEnabled` diatur ke `true`, DMS memerlukan Primary Key (PK) atau Unique Key (UK) pada tabel sumber. Tanpa PK atau Inggris pada tabel sumber, hanya sisipan batch yang diterapkan tetapi bukan pembaruan dan penghapusan batch.

Ketika `BatchApplyEnabled` diatur ke `true`, AWS DMS menghasilkan pesan kesalahan jika tabel target memiliki kendala unik dan kunci utama. Tabel target dengan kendala unik dan kunci primer tidak didukung saat `BatchApplyEnabled` disetel ke `true`.

Bila `BatchApplyEnabled` disetel ke `true` dan AWS DMS menemukan kesalahan data dari tabel dengan kebijakan penanganan kesalahan default, AWS DMS tugas akan beralih dari mode batch ke one-by-one mode untuk tabel lainnya. Untuk mengubah perilaku ini, Anda dapat mengatur tindakan "SUSPEND_TABLE" dalam kebijakan berikut dalam properti grup "ErrorBehavior" dari pengaturan tugas file JSON:

- `DataErrorPolicy`
- `ApplyErrorDeletePolicy`
- `ApplyErrorInsertPolicy`
- `ApplyErrorUpdatePolicy`

Untuk informasi lebih lanjut tentang "ErrorBehavior" properti grup ini, lihat contoh pengaturan tugas file JSON di [Menentukan setelan tugas untuk tugas AWS Database Migration Service](#). Setelah menyetel kebijakan ini "SUSPEND_TABLE", AWS DMS tugas kemudian menanggukkan kesalahan data pada tabel apa pun yang meningkatkannya dan melanjutkan dalam mode batch untuk semua tabel.

Anda dapat menggunakan parameter `BatchApplyEnabled` dengan parameter `BatchApplyPreserveTransaction`. Jika `BatchApplyEnabled` diatur ke `true`, maka `BatchApplyPreserveTransaction` parameter menentukan integritas transaksional.

Jika `BatchApplyPreserveTransaction` diatur ke `true`, maka integritas transaksional dipertahankan dan batch dijamin berisi semua perubahan dalam transaksi dari sumber.

Jika `BatchApplyPreserveTransaction` diatur ke `false`, maka bisa ada penyimpangan sementara dalam integritas transaksional untuk meningkatkan performa.

Parameter `BatchApplyPreserveTransaction` hanya berlaku untuk Oracle target akhir, dan hanya relevan ketika parameter `BatchApplyEnabled` diatur ke `true`.

Ketika LOB kolom termasuk dalam replikasi, Anda dapat menggunakan `BatchApplyEnabled` hanya dalam mode LOB terbatas.

Untuk informasi lebih lanjut tentang menggunakan pengaturan ini untuk beban change data capture (CDC), lihat [Mengubah pengaturan penyetelan pemrosesan](#).

- `MaxFullLoadSubTasks`— menunjukkan jumlah maksimum tabel untuk memuat secara paralel. Default adalah 8; nilai maksimum adalah 49.
- `ParallelLoadThreads`- Menentukan jumlah thread yang AWS DMS digunakan untuk memuat setiap tabel ke database target. Parameter ini memiliki nilai maksimum untuk target non-RDBMS. Nilai maksimum untuk target DynamoDB adalah 200. Nilai maksimum untuk target Amazon Kinesis Data Streams, Apache Kafka, atau OpenSearch Amazon Service adalah 32. Anda dapat meminta agar batas maksimum ini meningkat. `ParallelLoadThreads` berlaku untuk tugas Beban Penuh. Untuk informasi tentang pengaturan untuk beban paralel tabel individu, lihat [Tabel dan koleksi pengaturan aturan dan operasi](#).

Pengaturan ini berlaku untuk jenis engine endpoint berikut:

- DynamoDB
- Amazon Kinesis Data Streams
- Amazon MSK
- OpenSearch Layanan Amazon
- Amazon Redshift

AWS DMS mendukung `ParallelLoadThreads` untuk MySQL sebagai atribut koneksi tambahan. `ParallelLoadThread` tidak berlaku untuk MySQL sebagai pengaturan tugas.

- `ParallelLoadBufferSize` Menentukan jumlah maksimum catatan untuk disimpan dalam buffer yang digunakan thread beban paralel untuk memuat data ke target. Nilai default adalah 50. Nilai maksimumnya adalah 1.000. Pengaturan ini saat ini hanya berlaku ketika DynamoDB, Kinesis, Apache Kafka, atau adalah target. OpenSearch Gunakan parameter ini dengan `ParallelLoadThreads`. `ParallelLoadBufferSize` hanya berlaku bila ada lebih dari satu thread. Untuk informasi tentang pengaturan untuk beban paralel tabel individu, lihat [Tabel dan koleksi pengaturan aturan dan operasi](#).
- `ParallelLoadQueuesPerThread` – Menentukan jumlah antrian yang diakses setiap thread yang terjadi bersamaan untuk mengambil catatan data dari antrean dan menghasilkan beban batch untuk target. Default adalah 1. Pengaturan ini saat ini hanya berlaku bila Kinesis atau Apache Kafka menjadi targetnya.

- `ParallelApplyThreads`— Menentukan jumlah thread bersamaan yang AWS DMS digunakan selama pemuatan CDC untuk mendorong catatan data ke titik akhir target Amazon DocumentDB, Kinesis, Amazon MSK, atau Amazon Redshift. `OpenSearch Default` adalah nol (0).

Pengaturan ini hanya berlaku untuk CDC saja. Pengaturan ini tidak berlaku untuk Beban Penuh.

Pengaturan ini berlaku untuk jenis engine endpoint berikut:

- Amazon DocumentDB (dengan kompatibilitas MongoDB)
 - Amazon Kinesis Data Streams
 - Amazon Managed Streaming untuk Apache Kafka
 - OpenSearch Layanan Amazon
 - Amazon Redshift
- `ParallelApplyBufferSize`— Menentukan jumlah maksimum record yang akan disimpan di setiap antrian buffer untuk thread bersamaan untuk mendorong ke titik akhir target Amazon DocumentDB, Kinesis, Amazon MSK OpenSearch, atau Amazon Redshift selama pemuatan CDC. Nilai default-nya adalah 100. Nilai maksimumnya adalah 1000. Gunakan pilihan ini saat `ParallelApplyThreads` menentukan lebih dari satu thread.
 - `ParallelApplyQueuesPerThread`— Menentukan jumlah antrian yang diakses setiap thread untuk mengeluarkan catatan data dari antrian dan menghasilkan pemuatan batch untuk Amazon DocumentDB, Kinesis, Amazon MSK, atau endpoint selama CDC. `OpenSearch` Nilai default adalah 1.

Pengaturan tugas beban penuh

Pengaturan beban penuh meliputi yang berikut ini. Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

- Untuk menunjukkan bagaimana cara menangani pemuatan target pada startup beban penuh, tentukan salah satu nilai berikut untuk opsi `TargetTablePrepMode`:
 - `DO_NOTHING` – Data dan metadata dari tabel target yang ada tidak terpengaruh.
 - `DROP_AND_CREATE` – Tabel yang ada dijatuhkan dan tabel baru dibuat di tempatnya.
 - `TRUNCATE_BEFORE_LOAD` – Data terpotong tanpa memengaruhi metadata tabel.
- Untuk menunda pembuatan indeks kunci primer atau unik sampai setelah beban penuh selesai, atur opsi `CreatePkAfterFullLoad` untuk `true`.

- Untuk tugas beban penuh dan yang CDC-nya diaktifkan, Anda dapat mengatur opsi berikut untuk `Stop task after full load completes`:
 - `StopTaskCachedChangesApplied` – Atur opsi ini untuk `true` untuk menghentikan tugas setelah beban penuh selesai dan perubahan cache diterapkan.
 - `StopTaskCachedChangesNotApplied` – Atur opsi ini untuk `true` untuk menghentikan tugas sebelum perubahan cache diterapkan.
- Untuk menunjukkan jumlah maksimum tabel untuk memuat secara paralel, atur opsi `MaxFullLoadSubTasks`. Default adalah 8; nilai maksimum adalah 49.
- Tetapkan `ParallelLoadThreads` opsi untuk menunjukkan berapa banyak thread bersamaan yang akan digunakan DMS selama proses pemuatan penuh untuk mendorong catatan data ke titik akhir target. Nol adalah nilai default (0).

Important

`MaxFullLoadSubTasks` mengontrol jumlah tabel atau segmen tabel untuk memuat secara paralel. `ParallelLoadThreads` mengontrol jumlah thread yang digunakan oleh tugas migrasi untuk mengeksekusi beban secara paralel. Pengaturan ini bersifat multiplikatif. Dengan demikian, jumlah total utas yang digunakan selama tugas pemuatan penuh kira-kira merupakan hasil dari nilai **`ParallelLoadThreads`** dikalikan dengan nilai **`MaxFullLoadSubTasks`** (**`ParallelLoadThreads`** * `MaxFullLoadSubTasks`). Jika Anda membuat tugas dengan jumlah sub tugas Full Load yang tinggi dan sejumlah besar thread beban paralel, tugas Anda dapat menghabiskan terlalu banyak memori dan gagal.

- Anda dapat mengatur jumlah detik yang AWS DMS menunggu transaksi ditutup sebelum memulai operasi beban penuh. Untuk melakukannya, jika transaksi terbuka ketika tugas mulai atur opsi `TransactionConsistencyTimeout`. Nilai default adalah 600 (10 menit). AWS DMS memulai beban penuh setelah nilai batas waktu tercapai, bahkan jika ada transaksi terbuka. Sebuah full-load-only tugas tidak menunggu selama 10 menit tetapi segera dimulai.
- Untuk menunjukkan jumlah maksimum catatan yang dapat ditransfer bersama-sama, atur opsi `CommitRate`. Nilai default adalah 10000, dan nilai maksimum adalah 50000.

Pengaturan tugas Perjalanan Waktu

Untuk mencatat dan men-debug tugas replikasi, Anda dapat menggunakan AWS DMS Time Travel. Dalam pendekatan ini, Anda menggunakan Amazon S3 untuk menyimpan log dan mengenkripsi

mereka menggunakan kunci enkripsi Anda. Hanya dengan akses ke bucket Time Travel S3 Anda, Anda dapat mengambil log S3 Anda menggunakan filter tanggal-waktu, lalu melihat, mengunduh, dan mengaburkan log sesuai kebutuhan. Dengan melakukan ini, Anda dapat dengan aman “melakukan perjalanan kembali ke masa lalu” untuk menyelidiki aktivitas database. Time Travel bekerja secara independen dari CloudWatch logging. Untuk informasi lebih lanjut tentang CloudWatch logging, lihat [Pengaturan tugas pengelogan](#).

Anda dapat menggunakan Time Travel di semua AWS Wilayah dengan titik AWS DMS akhir sumber Oracle, Microsoft SQL Server, dan PostgreSQL yang didukung, dan titik akhir target PostgreSQL dan MySQL yang didukung. AWS DMS Anda dapat mengaktifkan Time Travel hanya untuk tugas full-load dan change data capture (CDC) dan untuk tugas khusus CDC. Untuk mengaktifkan Perjalanan Waktu atau mengubah pengaturan Perjalanan Waktu yang ada, pastikan tugas replikasi Anda dihentikan.

Pengaturan Perjalanan Waktu mencakup TTSettings properti berikut:

- **EnableTT**— Jika opsi ini diatur ke `true`, pencatatan Perjalanan Waktu diaktifkan untuk tugas tersebut. Nilai default-nya adalah `false`.

Tipe: Boolean

Wajib: Tidak

- **EncryptionMode**— Jenis enkripsi sisi server yang digunakan pada bucket S3 Anda untuk menyimpan data dan log Anda. Anda dapat menentukan `"SSE_S3"` (default) atau `"SSE_KMS"`.

Anda dapat mengubah `EncryptionMode` dari `"SSE_KMS"` ke `"SSE_S3"`, tetapi tidak sebaliknya.

Tipe: String

Wajib: Tidak

- **ServerSideEncryptionKmsKeyId**— Jika Anda menentukan `"SSE_KMS"` untuk `EncryptionMode`, berikan ID untuk AWS KMS kunci terkelola kustom Anda. Pastikan kunci yang Anda gunakan memiliki kebijakan terlampir yang mengaktifkan izin pengguna AWS Identity and Access Management (IAM) dan memungkinkan penggunaan kunci tersebut.

Hanya kunci KMS simetris yang dikelola khusus Anda sendiri yang didukung dengan opsi ini.

`"SSE_KMS"`

Jenis: String

Wajib: Hanya jika Anda menyetel `EncryptionMode` ke `"SSE_KMS"`

- `ServiceAccessRoleArn`— Nama Sumber Daya Amazon (ARN) yang digunakan oleh layanan untuk mengakses peran IAM. Tetapkan nama peran `kedms-tt-s3-access-role`. Ini adalah pengaturan wajib yang memungkinkan AWS DMS untuk menulis dan membaca objek dari ember S3.

Jenis: String

Wajib: Jika Perjalanan Waktu diaktifkan

Berikut ini adalah contoh kebijakan untuk peran ini.

```
{
  "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"
      ]
    }
  ]
}
```

Berikut ini adalah contoh kebijakan kepercayaan untuk peran ini.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
```

```

        "dms.amazonaws.com"
      ],
    },
    "Action": "sts:AssumeRole"
  }
]
}

```

- **BucketName**— Nama bucket S3 untuk menyimpan log Time Travel. Pastikan untuk membuat bucket S3 ini sebelum menyalakan log Time Travel.

Jenis: String

Wajib: Jika Perjalanan Waktu diaktifkan

- **BucketFolder**— Parameter opsional untuk mengatur nama folder di bucket S3. Jika Anda menentukan parameter ini, DMS membuat log Perjalanan Waktu di jalur `"/BucketName/BucketFolder/taskARN/YYYY/MM/DD/hh"`. Jika Anda tidak menentukan parameter ini, AWS DMS buat jalur default sebagai `"/BucketName/dms-time-travel-logs/taskARN/YYYY/MM/DD/hh"`.

Tipe: String

Wajib: Tidak

- **EnableDeletingFromS3OnTaskDelete**— Ketika opsi ini diatur ke `true`, AWS DMS menghapus log Perjalanan Waktu dari S3 jika tugas dihapus. Nilai default-nya adalah `false`.

Tipe: String

Wajib: Tidak

- **EnableRawData**— Ketika opsi ini diatur ke `true`, data mentah bahasa manipulasi data (DHTML) untuk log Perjalanan Waktu muncul di bawah `raw_data` kolom log Perjalanan Waktu. Untuk detailnya, lihat [Menggunakan log Perjalanan Waktu](#). Nilai default-nya adalah `false`. Ketika opsi ini diatur ke `false`, hanya jenis DMLnya yang ditangkap.

Tipe: String

Wajib: Tidak

- **RawDataFormat**— Dalam AWS DMS versi 3.5.0 dan lebih tinggi, ketika `EnableRawData` diatur ke `true`. Properti ini menentukan format untuk data mentah DML dalam log Time Travel dan dapat disajikan sebagai:

- "TEXT"— Nama dan nilai kolom yang diuraikan dan dapat dibaca untuk peristiwa DHTML yang ditangkap selama CDC sebagai bidang. Raw
- "HEX"— Hexidecimal asli untuk nama kolom dan nilai yang ditangkap untuk peristiwa DMLselama CDC.

Properti ini berlaku untuk sumber database Oracle dan Microsoft SQL Server.

Tipe: String

Wajib: Tidak

- `OperationsToLog`- Menentukan jenis operasi DML untuk login log Time Travel. Anda dapat menentukan salah satu hal berikut:
 - "INSERT"
 - "UPDATE"
 - "DELETE"
 - "COMMIT"
 - "ROLLBACK"
 - "ALL"

Default-nya adalah "ALL".

Tipe: String

Wajib: Tidak

- `MaxRecordSize`— Menentukan ukuran maksimum catatan log Perjalanan Waktu yang dicatat untuk setiap baris. Gunakan properti ini untuk mengontrol pertumbuhan log Perjalanan Waktu untuk tabel yang sangat sibuk. Default adalah 64 KB.

Tipe: Integer

Wajib: Tidak

Untuk informasi selengkapnya tentang mengaktifkan dan menggunakan log Perjalanan Waktu, lihat topik berikut.

Topik

- [Mengaktifkan log Perjalanan Waktu untuk suatu tugas](#)

- [Menggunakan log Perjalanan Waktu](#)
- [Seberapa sering AWS DMS mengunggah log Time Travel ke S3](#)

Mengaktifkan log Perjalanan Waktu untuk suatu tugas

Anda dapat mengaktifkan Perjalanan Waktu untuk AWS DMS tugas menggunakan pengaturan tugas yang dijelaskan sebelumnya. Pastikan tugas replikasi Anda dihentikan sebelum Anda mengaktifkan Perjalanan Waktu.

Untuk mengaktifkan Perjalanan Waktu menggunakan AWS CLI

1. Buat file JSON konfigurasi tugas DMS dan tambahkan TTSettings bagian seperti berikut ini. Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

```
.
.
.
  },
  "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. Dalam tindakan tugas yang sesuai, tentukan file JSON ini menggunakan `--replication-task-settings` opsi. Misalnya, fragmen kode CLI berikut menentukan file pengaturan Time Travel ini sebagai bagian dari `create-replication-task`

```
aws dms create-replication-task
--target-endpoint-arn arn:aws:dms:us-
east-1:112233445566:endpoint:ELS507YTYV452CAZR2EYBNQGILFHQIFVPWFRQAY \
--source-endpoint-arn arn:aws:dms:us-
east-1:112233445566:endpoint:HNX2BWIIN5ZYFF7F6UFFZVWTDFFSMTN0V2FTXZA \
--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
.
.
.
```

Di sini, nama file pengaturan Perjalanan Waktu ini adalah `task-settings-tt-enabled.json`.

Demikian pula, Anda dapat menentukan file ini sebagai bagian dari `modify-replication-task` tindakan.

Perhatikan penanganan khusus log Perjalanan Waktu untuk tindakan tugas berikut:

- `start-replication-task`— Saat Anda menjalankan tugas replikasi, jika bucket S3 yang digunakan untuk Perjalanan Waktu tidak dapat diakses, tugas tersebut ditandai sebagai `FAILED`
- `stop-replication-task`— Saat tugas berhenti, AWS DMS segera dorong semua log Perjalanan Waktu yang saat ini tersedia untuk instance replikasi ke bucket S3 yang digunakan untuk Perjalanan Waktu.

Saat tugas replikasi berjalan, Anda dapat mengubah `EncryptionMode` nilai dari `"SSE_KMS"` ke `"SSE_S3"` tetapi tidak sebaliknya.

Jika ukuran log Time Travel untuk tugas yang sedang berlangsung melebihi 1 GB, DMS mendorong log ke S3 dalam waktu lima menit setelah mencapai ukuran itu. Setelah tugas berjalan, jika bucket S3

atau kunci KMS menjadi tidak dapat diakses, DMS berhenti mendorong log ke bucket ini. Jika Anda menemukan log Anda tidak didorong ke bucket S3 Anda, periksa S3 dan AWS KMS izin Anda. Untuk detail lebih lanjut tentang seberapa sering DMS mendorong log ini ke S3, lihat. [Seberapa sering AWS DMS mengunggah log Time Travel ke S3](#)

Untuk mengaktifkan Perjalanan Waktu untuk tugas yang ada dari konsol, gunakan opsi editor JSON di bawah Pengaturan Tugas untuk menambahkan TTSettings bagian.

Menggunakan log Perjalanan Waktu

File log Perjalanan Waktu adalah file nilai dipisahkan koma (CSV) dengan bidang berikut.

```
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
```

Setelah log Perjalanan Waktu Anda tersedia di S3, Anda dapat langsung mengakses dan menanyakannya dengan alat seperti Amazon Athena. Atau Anda dapat mengunduh log sebagaimana Anda dapat file apa pun dari S3.

Contoh berikut menunjukkan log Time Travel di mana transaksi untuk tabel yang mytable disebut dicatat. Akhiran baris untuk log berikut ditambahkan untuk keterbacaan.

```
"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'"
```

Seberapa sering AWS DMS mengunggah log Time Travel ke S3

Untuk meminimalkan penggunaan penyimpanan instans replikasi Anda, AWS DMS bongkar log Perjalanan Waktu darinya secara berkala.

Log perjalanan waktu didorong ke bucket Amazon S3 Anda dalam kasus berikut:

- Jika ukuran log saat ini melebihi 1 GB, AWS DMS unggah log ke S3 dalam waktu lima menit. Dengan demikian, AWS DMS dapat membuat hingga 12 panggilan per jam ke S3 dan AWS KMS untuk setiap tugas yang berjalan.
- AWS DMS mengunggah log ke S3 setiap jam, terlepas dari ukuran log.
- Saat tugas dihentikan, AWS DMS segera unggah log perjalanan waktu ke S3.

Pengaturan tugas pengelolan

Logging menggunakan Amazon CloudWatch untuk mencatat informasi selama proses migrasi. Menggunakan pengaturan tugas pencatatan, Anda dapat menentukan kegiatan komponen yang dicatat dan jumlah informasi yang ditulis ke log. Pengaturan tugas penebangan ditulis ke file JSON. Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

Anda dapat mengaktifkan CloudWatch logging dengan beberapa cara. Anda dapat memilih EnableLogging opsi pada AWS Management Console saat Anda membuat tugas migrasi. Atau, Anda dapat mengatur EnableLogging opsi true saat membuat tugas menggunakan AWS DMS API. Anda juga dapat menentukan "EnableLogging": true di JSON bagian logging pengaturan tugas.

Saat Anda menyetel EnableLogging ketru, AWS DMS tetapkan nama CloudWatch grup dan nama aliran sebagai berikut. Anda tidak dapat mengatur nilai-nilai ini secara langsung.

- CloudWatchLogGroup: dms-tasks-<REPLICATION_INSTANCE_IDENTIFIER>
- CloudWatchLogStream: dms-task-<REPLICATION_TASK_EXTERNAL_RESOURCE_ID>

<REPLICATION_INSTANCE_IDENTIFIER> adalah pengenal dari contoh replikasi.
<REPLICATION_TASK_EXTERNAL_RESOURCE_ID> adalah nilai dari <resourcename> bagian Tugas ARN. Untuk informasi tentang cara AWS DMS menghasilkan ARN sumber daya, lihat [Membangun Nama Sumber Daya Amazon \(ARN\) untuk AWS DMS](#).

CloudWatch terintegrasi dengan AWS Identity and Access Management (IAM), dan Anda dapat menentukan CloudWatch tindakan mana yang dapat dilakukan pengguna di AWS akun Anda. Untuk informasi selengkapnya tentang bekerja dengan IAM CloudWatch, lihat [Identitas dan manajemen akses untuk amazon CloudWatch](#) dan [Pencatatan panggilan CloudWatch API Amazon](#) di Panduan CloudWatch Pengguna Amazon.

Untuk menghapus log tugas, Anda dapat mengatur DeleteTaskLogs ke betul di JSON bagian logging dari pengaturan tugas.

Anda dapat menentukan pencatatan untuk jenis peristiwa berikut:

- FILE_FACTORY— Pabrik file mengelola file yang digunakan untuk penerapan batch dan pemuatan batch, dan mengelola titik akhir Amazon S3.
- METADATA_MANAGER— Manajer metadata mengelola metadata sumber dan target, partisi, dan status tabel selama replikasi.
- SORTER— SORTER Menerima peristiwa yang masuk dari SOURCE_CAPTURE proses. Peristiwa dikumpulkan dalam transaksi, dan diteruskan ke komponen TARGET_APPLY layanan. Jika SOURCE_CAPTURE proses menghasilkan peristiwa lebih cepat daripada yang dapat dikonsumsi TARGET_APPLY komponen, SORTER komponen menyimpan peristiwa yang di-backlog ke disk atau ke file swap. Peristiwa yang di-cache adalah penyebab umum kehabisan penyimpanan dalam contoh replikasi.

Komponen SORTER layanan mengelola peristiwa yang di-cache, mengumpulkan statistik CDC, dan melaporkan latensi tugas.

- SOURCE_CAPTURE— Data replikasi berkelanjutan (CDC) diambil dari database sumber atau layanan, dan diteruskan ke komponen layanan SORTER.
- SOURCE_UNLOAD— Data diturunkan dari database sumber atau layanan selama Beban Penuh.
- TABLES_MANAGER— Manajer tabel melacak tabel yang diambil, mengelola urutan migrasi tabel, dan mengumpulkan statistik tabel.
- TARGET_APPLY— Pernyataan bahasa definisi data dan data (DDL) diterapkan ke database target.
- TARGET_LOAD— Data dimuat ke database target.

- **TASK_MANAGER**—Pengelola tugas mengelola tugas yang sedang berjalan, dan memecah tugas menjadi sub-tugas untuk pemrosesan data paralel.
- **TRANSFORMATION**— Acara transformasi pemetaan tabel. Untuk informasi selengkapnya, lihat [Menggunakan pemetaan tabel untuk menentukan pengaturan tugas](#).
- **VALIDATOR/ VALIDATOR_EXT**—Komponen VALIDATOR layanan memverifikasi bahwa data dimigrasi secara akurat dari sumber ke target. Untuk informasi selengkapnya, lihat [Validasi data](#).

Komponen logging berikut menghasilkan sejumlah besar log saat menggunakan tingkat keparahan **LOGGER_SEVERITY_DETAILED_DEBUG** log:

- COMMON
- ADDONS
- DATA_STRUCTURE
- COMMUNICATION
- FILE_TRANSFER
- FILE_FACTORY

Level logging selain **DEFAULT** jarang diperlukan untuk komponen ini selama pemecahan masalah. Kami tidak menyarankan untuk mengubah tingkat logging dari komponen ini **DEFAULT** kecuali diminta secara khusus oleh AWS Support.

Setelah Anda menentukan salah satu sebelumnya, Anda kemudian dapat menentukan jumlah informasi yang dicatat, seperti yang ditunjukkan dalam daftar berikut.

Tingkat keparahan adalah dalam urutan dari terendah ke tingkat tertinggi informasi. Tingkat yang lebih tinggi selalu mencakup informasi dari tingkat yang lebih rendah.

- **LOGGER_SEVERITY_ERROR** - pesan kesalahan ditulis ke log.
- **LOGGER_SEVERITY_WARNING** - Peringatan dan pesan kesalahan ditulis ke log.
- **LOGGER_SEVERITY_INFO** -informasi pesan, peringatan, dan pesan kesalahan ditulis ke log.
- **LOGGER_SEVERITY_DEFAULT** - pesan informasi, peringatan, dan pesan kesalahan ditulis ke log.
- **LOGGER_SEVERITY_DEBUG** - pesan Debug, pesan informasi, peringatan, dan pesan kesalahan ditulis ke log.
- **LOGGER_SEVERITY_DETAILED_DEBUG** - Semua informasi ditulis ke log.

Contoh JSON berikut menunjukkan pengaturan tugas untuk log semua tindakan dan tingkat kepelikan.

```
...
  "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
},
...
```

Mengontrol pengaturan tugas tabel

Tabel kontrol memberikan informasi tentang AWS DMS tugas. Mereka juga menyediakan statistik berguna yang dapat Anda gunakan untuk merencanakan dan mengelola tugas migrasi saat ini dan tugas di masa mendatang. Anda dapat menerapkan pengaturan tugas ini dalam file JSON atau dengan memilih Pengaturan Lanjutan pada halaman Buat tugas di AWS DMS konsol. Tabel Terapkan Pengecualian (`dmslogs.aws_dms_apply_exceptions`) selalu dibuat pada target database. Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

AWS DMS hanya membuat tabel kontrol hanya selama tugas Full Load+CDC atau CDC saja, dan tidak selama tugas Full Load Only.

Untuk beban penuh dan tugas CDC (Migrasi data yang ada dan mereplikasi perubahan yang sedang berlangsung) dan hanya CDC (Replikasi perubahan data saja), Anda juga dapat membuat tabel tambahan, termasuk yang berikut ini:

- Status replikasi (`dmslogs.aws_dms_status`) – Tabel ini memberikan detail tentang tugas saat ini. Ini termasuk status tugas, jumlah memori yang dikonsumsi oleh tugas, dan jumlah perubahan belum diterapkan ke target. Tabel ini juga memberikan posisi dalam database sumber tempat AWS DMS sedang membaca. Juga, ini menunjukkan jika tugas dalam fase beban penuh atau perubahan data capture (CDC).
- Tabel Ditangguhkan (`dmslogs.aws_dms_suspended_tables`) – Tabel ini menyediakan daftar tabel yang ditangguhkan serta alasan tabel tersebut ditangguhkan.
- Riwayat replikasi (`dmslogs.aws_dms_history`) – Tabel ini menyediakan informasi tentang riwayat replikasi. Informasi ini mencakup jumlah dan volume catatan diproses selama tugas, latensi pada akhir tugas CDC, dan statistik lainnya.

Tabel Terapkan Pengecualian (`dmslogs.aws_dms_apply_exceptions`) berisi parameter-parameter berikut ini.

Kolom	Jenis	Deskripsi
TASK_NAME	nvchar	ID Sumber Daya AWS DMS tugas. ID sumber daya dapat ditemukan di tugas ARN.
TABLE_OWNER	nvchar	Pemilik meja.
table_name	nvchar	Nama tabel.
ERROR_TIME	timestamp	Waktu pengecualian (kesalahan) terjadi.
PERNYATAAN	nvchar	Pernyataan yang sedang dijalankan ketika kesalahan terjadi.
Kesalahan	nvchar	Nama kesalahan dan deskripsi.

Tabel Status replikasi (`dmslogs.aws_dms_status`) berisi status tugas dan basis data target. Ini memiliki pengaturan berikut.

Kolom	Jenis	Deskripsi
SERVER_NAME	nvchar	Nama mesin di mana tugas replikasi berjalan.
TASK_NAME	nvchar	ID Sumber Daya AWS DMS tugas. ID sumber daya dapat ditemukan di tugas ARN.
TASK_STATUS	varchar	Salah satu nilai berikut: <ul style="list-style-type: none"> • BEBAN PENUH • MENGUBAH PEMROSESAN (CDC) • TIDAK BERJALAN

Kolom	Jenis	Deskripsi
		Status tugas diatur ke LOAD LENGKAP selama ada setidaknya satu meja dalam beban penuh. Setelah semua tabel telah dimuat, perubahan status tugas untuk mengubah PROSES jika CDC diaktifkan. Tugas diatur ke TIDAK BERJALAN sebelum Anda memulai tugas, atau setelah tugas selesai.
STATUS_TIME	timestamp	Timestamp dari status tugas.
PENDING_CHANGES	int	Jumlah catatan perubahan yang dilakukan dalam database sumber dan di-cache dalam memori dan disk instance replikasi Anda.
DISK_SWAP_SIZE	int	Jumlah ruang disk yang digunakan oleh transaksi lama atau offloaded.
TASK_MEMORY	int	Memori saat ini digunakan, dalam MB.
SOURCE_CURRENT _POSISI	varchar	Posisi dalam database sumber yang saat AWS DMS ini sedang membaca dari.
SOURCE_CURRENT _TIMESTAMP	timestamp	Stempel waktu dalam database sumber yang saat ini AWS DMS sedang membaca dari.
SOURCE_TAIL _POSISI	varchar	Posisi transaksi awal tertua yang tidak berkomitmen. Nilai ini adalah posisi terbaru yang dapat Anda kembalikan tanpa kehilangan perubahan apa pun.

Kolom	Jenis	Deskripsi
SOURCE_TAIL_TIMESTAMP	timestamp	Cap waktu transaksi awal tertua yang tidak berkomitmen. Nilai ini adalah stempel waktu terbaru yang dapat Anda kembalikan tanpa kehilangan perubahan apa pun.
SOURCE_TIMESTAMP_DITERAPKAN	timestamp	Timestamp dari komit transaksi terakhir. Dalam sebagian besar menerapkan proses, nilai ini adalah timestamp untuk komit transaksi terakhir dalam batch.

Tabel Suspended (`dmslogs.aws_dms_suspended_tables`) berisi parameter berikut.

Kolom	Jenis	Deskripsi
SERVER_NAME	nvarchar	Nama mesin di mana tugas replikasi berjalan.
TASK_NAME	nvarchar	Nama AWS DMS tugas
TABLE_OWNER	nvarchar	Pemilik meja.
table_name	nvarchar	Nama tabel.
SUSPEND_ALASAN	nvarchar	Alasan penangguhan.
SUSPEND_TIMESTAMP	timestamp	Waktu penangguhan terjadi.

Tabel sejarah replikasi (`dmslogs.aws_dms_history`) berisi parameter-parameter berikut ini.

Kolom	Jenis	Deskripsi
SERVER_NAME	nvarchar	Nama mesin di mana tugas replikasi berjalan.
TASK_NAME	nvarchar	ID Sumber Daya AWS DMS tugas. ID sumber daya dapat ditemukan di tugas ARN.
TIMESLOT_TYPE	varchar	Salah satu nilai berikut: <ul style="list-style-type: none"> • BEBAN PENUH • MENGUBAH PEMROSESAN (CDC) <p>Jika tugas menjalankan beban penuh dan CDC, dua catatan sejarah ditulis ke slot waktu.</p>
TIMESLOT	timestamp	Timestamp berakhir dari slot waktu.
TIMESLOT_DURATION	int	Durasi slot waktu, dalam hitungan menit.
TIMESLOT_LATENCY	int	Latensi target pada akhir slot waktu, dalam hitungan detik. Nilai ini hanya berlaku untuk slot waktu CDC.
CATATAN	int	Jumlah catatan diproses selama slot waktu.
TIMESLOT_VOLUME	int	Volume data yang diproses dalam MB.

Tabel Kegagalan Validasi (`awsdms_validation_failures_v1`) berisi semua kegagalan validasi data untuk tugas. Untuk informasi selengkapnya lihat, Pemecahan [Masalah Validasi Data](#).

Pengaturan tabel kontrol tambahan mencakup hal berikut hal berikut:

- `HistoryTimeslotInMinutes` – Gunakan opsi ini untuk menunjukkan panjang setiap slot waktu dalam tabel Riwayat Replikasi. Default adalah 5 menit.
- `ControlSchema`— Gunakan opsi ini untuk menunjukkan nama skema database untuk tabel kontrol untuk AWS DMS target. Jika Anda tidak memasukkan informasi apapun untuk pilihan ini, maka tabel disalin ke lokasi default dalam basis data seperti yang tercantum berikut:
 - PostgreSQL, Publik
 - Oracle, skema target
 - Microsoft SQL Server, dbo dalam basis data target
 - MySQL, `awsdms_control`
 - MariaDB, `awsdms_control`
 - Amazon Redshift, Publik
 - DynamoDB, dibuat sebagai tabel individu dalam basis data
 - IBM Db2 LUW, `awsdms_kontrol`

Pengaturan tugas buffer pengaliran

Anda dapat mengatur pengaturan buffer streaming menggunakan AWS CLI, termasuk yang berikut ini. Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

- `StreamBufferCount` – Gunakan opsi ini untuk menentukan jumlah buffer aliran data untuk tugas migrasi. Default jumlah buffer pengaliran adalah 3. Meningkatkan nilai pengaturan ini dapat meningkatkan kecepatan ekstraksi data. Namun, peningkatan performa ini sangat tergantung pada lingkungan migrasi, termasuk sistem sumber dan kelas instans server replikasi. Default cukup untuk sebagian besar situasi.
- `StreamBufferSizeInMB` – Gunakan opsi ini untuk menunjukkan ukuran maksimum setiap buffer aliran data. Ukuran default adalah 8 MB. Anda mungkin perlu meningkatkan nilai untuk opsi ini ketika Anda bekerja dengan LOB yang sangat besar. Anda juga mungkin perlu meningkatkan nilai jika Anda menerima pesan dalam berkas log yang ukuran buffer pengaliran tidak mencukupi. Saat menghitung ukuran opsi ini, Anda dapat menggunakan persamaan berikut: $[\text{Max LOB size (or LOB chunk size)}] * [\text{number of LOB}$

`columns]*[number of stream buffers]*[number of tables loading in parallel per task(MaxFullLoadSubTasks)]*3`

- `CtrlStreamBufferSizeInMB` — Gunakan opsi ini untuk mengatur ukuran buffer pengaliran kontrol. Nilai dalam megabyte, dan dapat 1-8. Nilai default adalah 5. Anda mungkin perlu meningkatkan ini saat bekerja dengan jumlah tabel yang sangat besar, seperti puluhan ribu tabel.

Mengubah pengaturan penyetelan pemrosesan

Pengaturan berikut menentukan cara AWS DMS menangani perubahan untuk tabel target selama pengambilan data perubahan (CDC). Beberapa pengaturan ini bergantung pada nilai parameter metadata target `BatchApplyEnabled`. Untuk informasi lebih lanjut tentang parameter `BatchApplyEnabled`, lihat [Menargetkan pengaturan tugas metadata](#). Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

Pengaturan penyetelan pemrosesan yang benar mencakup hal berikut:

Pengaturan berikut hanya berlaku bila parameter metadata target `BatchApplyEnabled` diatur ke `true`.

- `BatchApplyPreserveTransaction` – Jika diatur ke `true`, integritas transaksional dipertahankan dan batch dijamin berisi semua perubahan dalam transaksi dari sumber. Nilai default-nya adalah `true`. Pengaturan ini hanya berlaku untuk Oracle target endpoint.

Jika diatur ke `false`, dapat ada penyimpangan sementara dalam integritas transaksional untuk meningkatkan performa. Tidak ada jaminan bahwa semua perubahan dalam transaksi dari sumber diterapkan ke target dalam batch tunggal.

Secara default, AWS DMS proses berubah dalam mode transaksional, yang menjaga integritas transaksional. Jika Anda mampu penyimpangan sementara dalam integritas transaksional, Anda dapat menggunakan batch dioptimalkan menerapkan pilihan sebagai gantinya. Opsi ini secara efisien mengelompokkan transaksi dan menerapkannya dalam batch untuk tujuan efisiensi. Menggunakan batch dioptimalkan menerapkan opsi hampir selalu melanggar referensial integritas kendala. Jadi kami sarankan Anda menonaktifkan kendala ini selama proses migrasi dan mengaktifkannya lagi sebagai bagian dari proses cutover.

- `BatchApplyTimeoutMin`— Menetapkan jumlah waktu minimum dalam detik yang AWS DMS menunggu di antara setiap aplikasi perubahan batch. Nilai default adalah 1.

- `BatchApplyTimeoutMax`— Mengatur jumlah waktu maksimum dalam detik yang AWS DMS menunggu di antara setiap aplikasi perubahan batch sebelum waktu habis. Nilai default-nya adalah 30.
- `BatchApplyMemoryLimit` – Menetapkan jumlah maksimum memori dalam (MB) untuk digunakan untuk pra-pemrosesan di Mode penerapan yang dioptimalkan batch. Nilai default adalah 500.
- `BatchSplitSize` – Mengatur jumlah maksimum perubahan yang diterapkan dalam satu batch. Nilai default 0, yang berarti tidak ada batas diterapkan.

Pengaturan berikut hanya berlaku bila parameter metadata target `BatchApplyEnabled` diatur ke `false`.

- `MinTransactionSize` – Menetapkan jumlah minimum perubahan untuk dimasukkan dalam setiap transaksi. Nilai default adalah 1000.
- `CommitTimeout`— Menetapkan waktu maksimum dalam hitungan detik AWS DMS untuk mengumpulkan transaksi dalam batch sebelum menyatakan batas waktu. Nilai default adalah 1.

Untuk replikasi dua arah, pengaturan berikut hanya berlaku ketika parameter metadata target disetel ke `BatchApplyEnabled false`

- `LoopbackPreventionSettings` – Pengaturan ini memberikan pencegahan loopback untuk setiap tugas replikasi yang sedang berlangsung dalam setiap pasangan tugas yang terlibat dalam replikasi dua arah. Pencegahan Loopback mencegah perubahan identik diterapkan di kedua arah replikasi dua arah, yang dapat merusak data. Untuk informasi selengkapnya tentang replikasi dua arah, lihat [Melakukan replikasi dua arah](#).

AWS DMS upaya untuk menyimpan data transaksi dalam memori sampai transaksi sepenuhnya berkomitmen pada sumber, target, atau keduanya. Namun, transaksi yang lebih besar dari memori dialokasikan atau yang tidak berkomitmen dalam batas waktu tertentu ditulis ke disk.

Pengaturan berikut berlaku untuk mengubah penyetelan pemrosesan terlepas dari perubahan mode pengolahan.

- `MemoryLimitTotal` – Menetapkan ukuran maksimum (dalam MB) yang dapat ditempati semua transaksi dalam memori sebelum ditulis ke disk. Nilai default adalah 1024.

- **MemoryKeepTime** – Mengatur waktu maksimum dalam detik di mana setiap transaksi dapat tinggal dalam memori sebelum ditulis ke disk. Durasi dihitung dari waktu yang AWS DMS mulai menangkap transaksi. Nilai defaultnya adalah 60.
- **StatementCacheSize** – Menetapkan jumlah maksimum pernyataan yang disiapkan untuk disimpan di server untuk dieksekusi di masa depan ketika menerapkan perubahan ke target. Nilai default adalah 50. Nilai maksimumnya adalah 200 tahun.

Berikut ini adalah contoh bagaimana pengaturan tugas yang menangani Change Processing Tuning muncul dalam file JSON pengaturan tugas:

```
"ChangeProcessingTuning": {
  "BatchApplyPreserveTransaction": true,
  "BatchApplyTimeoutMin": 1,
  "BatchApplyTimeoutMax": 30,
  "BatchApplyMemoryLimit": 500,
  "BatchSplitSize": 0,
  "MinTransactionSize": 1000,
  "CommitTimeout": 1,
  "MemoryLimitTotal": 1024,
  "MemoryKeepTime": 60,
  "StatementCacheSize": 50
}
```

Untuk mengontrol frekuensi menulis ke target Amazon S3 selama tugas replikasi data, Anda dapat mengonfigurasi atribut koneksi tambahan `cdcMaxBatchInterval` dan `cdcMinFileSize`. Hal ini dapat menghasilkan performa yang lebih baik ketika menganalisis data tanpa operasi overhead tambahan. Untuk informasi selengkapnya, lihat [Pengaturan titik akhir saat menggunakan Amazon S3 sebagai target untuk AWS DMS](#).

Pengaturan tugas validasi data

Anda dapat memastikan bahwa data Anda dimigrasi secara akurat dari sumber ke target. Jika Anda mengaktifkan validasi untuk tugas, AWS DMS mulailah membandingkan sumber dan data target segera setelah beban penuh dilakukan untuk tabel. Untuk informasi lebih lanjut tentang validasi data tugas, persyaratan, cakupan dukungan basis data, dan metrik yang dilaporkan, lihat [Validasi data AWS DMS](#). Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

Pengaturan validasi data dan nilai-nilai mereka mencakup hal berikut:

- `EnableValidation` – Mengaktifkan validasi data bila diatur ke `BETUL`. Jika tidak, validasi dinonaktifkan untuk tugas tersebut. Nilai default-nya adalah `SALAH`.
- `ValidationMode` – Mengontrol bagaimana DMS akan memvalidasi data dalam tabel target terhadap tabel sumber. AWS DMS menyediakan pengaturan ini untuk perpanjangan di masa mendatang. Saat ini, nilai default dan satu-satunya yang valid adalah `ROW_LEVEL`. AWS DMS memvalidasi semua baris antara tabel sumber dan target.
- `FailureMaxCount` – Menentukan jumlah maksimum catatan yang dapat gagal validasi sebelum validasi ditangguhkan untuk tugas tersebut. Nilai default adalah 10,000. Jika Anda ingin validasi untuk melanjutkan terlepas dari jumlah catatan yang gagal validasi, menetapkan nilai ini lebih tinggi dari jumlah catatan dalam sumber.
- `HandleCollationDiff`— Ketika opsi ini diatur ke `true`, validasi memperhitungkan perbedaan pengumpulan kolom di PostgreSQL dan titik akhir Microsoft SQL Server saat mengidentifikasi sumber dan catatan target untuk dibandingkan. Jika tidak, perbedaan tersebut dalam pemeriksaan kolom diabaikan untuk validasi. Collations kolom dapat mendikte urutan baris, yang penting untuk validasi data. Mengatur `HandleCollationDiff` ke `BENAR` menyelesaikan perbedaan kolasi secara otomatis dan mencegah positif palsu dalam validasi data. Nilai default-nya adalah `false`.
- `RecordFailureDelayInMinutes`— Menentukan waktu tunda dalam beberapa menit sebelum melaporkan rincian kegagalan validasi.
- `RecordFailureDelayLimitInMinutes` – Menentukan penundaan sebelum melaporkan detail kegagalan validasi. Biasanya, AWS DMS menggunakan latensi tugas untuk mengenali penundaan aktual agar perubahan sampai ke target untuk mencegah positif palsu. Pengaturan ini mengesampingkan nilai penundaan aktual dan memungkinkan Anda mengatur penundaan yang lebih tinggi sebelum melaporkan metrik validasi apa pun. Nilai default adalah 0.
- `RecordSuspendDelayInMinutes`- Menentukan waktu tunda dalam beberapa menit sebelum tabel ditangguhkan dari validasi karena ambang kesalahan ditetapkan. `FailureMaxCount`
- `SkipLobColumns`— Ketika opsi ini diatur ke `true`, AWS DMS lewati validasi data untuk semua kolom LOB di bagian tabel dari validasi tugas. Nilai default-nya adalah `false`.
- `TableFailureMaxCount`- Menentukan jumlah maksimum baris dalam satu tabel yang dapat gagal validasi sebelum validasi ditangguhkan untuk tabel. Nilai default adalah 1,000.
- `ThreadCount`— Menentukan jumlah thread eksekusi yang AWS DMS menggunakan selama validasi. Setiap thread memilih not-yet-validated data dari sumber dan target untuk membandingkan dan memvalidasi. Nilai bawaannya adalah 5. Jika Anda mengatur `ThreadCount` ke angka yang lebih tinggi, AWS DMS dapat menyelesaikan validasi lebih cepat. Namun, AWS

DMS kemudian menjalankan kueri yang lebih simultan, mengkonsumsi lebih banyak sumber daya pada sumber dan target.

- `ValidationOnly`— Ketika opsi ini diatur ke `true`, tugas melakukan validasi data tanpa melakukan migrasi atau replikasi data apa pun. Nilai default-nya adalah `false`. Anda tidak dapat mengubah `ValidationOnly` pengaturan setelah tugas dibuat.

Anda harus mengatur `TargetTablePrepMode` ke `DO_NOTHING` (default untuk tugas validasi saja) dan menyetel Jenis Migrasi ke salah satu dari berikut ini:

- `Full Load` — Setel tugas Jenis migrasi ke Migrasi data yang ada di AWS DMS konsol. Atau, di AWS DMS API, atur jenis migrasi ke `FULL-LOAD`.
- `CDC` — Setel tugas Jenis migrasi untuk Mereplikasi perubahan data hanya di konsol. AWS DMS Atau, di AWS DMS API, atur jenis migrasi ke `CDC`.

Terlepas dari jenis migrasi yang dipilih, data sebenarnya tidak dimigrasikan atau direplikasi selama tugas validasi saja.

Untuk informasi selengkapnya, lihat [Tugas hanya validasi](#).

 Important

Parameter `ValidationOnly` tetap. Itu tidak dapat dimodifikasi untuk tugas setelah tugas itu dibuat.

- `ValidationPartialLobSize` – Menentukan jika Anda ingin melakukan validasi parsial untuk kolom LOB alih-alih memvalidasi semua data yang disimpan dalam kolom. Ini adalah sesuatu yang Anda mungkin menemukan berguna ketika Anda bermigrasi hanya bagian dari data LOB dan tidak seluruh LOB data set. Nilai dalam unit KB. Nilai default-nya adalah 0, yang berarti AWS DMS memvalidasi semua data kolom LOB. Misalnya, "`ValidationPartialLobSize`": 32 berarti AWS DMS hanya memvalidasi 32KB pertama dari data kolom di sumber dan target.
- `PartitionSize` – Menentukan ukuran batch catatan untuk dibaca sebagai perbandingan dari sumber dan target. Default adalah 10.000.
- `ValidationQueryCdcDelaySeconds` – Jumlah waktu kueri validasi pertama tertunda pada sumber dan target untuk setiap pembaruan CDC. Hal ini dapat membantu mengurangi pertentangan sumber daya ketika latensi migrasi tinggi. Sebuah validasi hanya tugas secara otomatis menetapkan opsi ini untuk 180 detik. Defaultnya adalah 0.

Misalnya, JSON berikut memungkinkan validasi data dengan dua kali jumlah default benang. Ini juga menyumbang perbedaan dalam urutan catatan yang disebabkan oleh perbedaan pemeriksaan kolom di endpoint PostgreSQL. Selain itu, menyediakan penundaan pelaporan validasi ke akun untuk waktu tambahan untuk memproses kegagalan validasi.

```
"ValidationSettings": {
  "EnableValidation": true,
  "ThreadCount": 10,
  "HandleCollationDiff": true,
  "RecordFailureDelayLimitInMinutes": 30
}
```

Note

Untuk endpoint Oracle, AWS DMS gunakan DBMS_CCRYPTO untuk memvalidasi BLOB. Jika endpoint Oracle Anda menggunakan BLOB, berikan execute izin untuk DBMS_CCRYPTO ke akun pengguna yang mengakses endpoint Oracle. Untuk melakukan ini, jalankan pernyataan berikut.

```
grant execute on sys.dbms_crypto to dms_endpoint_user;
```

Pengaturan tugas untuk perubahan pengolahan DDL penanganan

Pengaturan berikut menentukan bagaimana AWS DMS menangani perubahan bahasa definisi data (DDL) untuk tabel target selama pengambilan data perubahan (CDC). Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

Pengaturan tugas untuk menangani perubahan pengolahan DDL meliputi berikut ini:

- `HandleSourceTableDropped` – Atur opsi ini ke `true` untuk menjatuhkan tabel target ketika tabel sumber dijatuhkan.
- `HandleSourceTableTruncated` – Atur opsi ini ke `true` untuk memotong tabel target ketika tabel sumber dipotong.
- `HandleSourceTableAltered` – Atur opsi ini ke `true` untuk mengubah tabel target ketika tabel sumber diubah.

Berikut ini adalah contoh bagaimana pengaturan tugas yang menangani perubahan pengolahan DDL muncul dalam tugas pengaturan file JSON:

```
"ChangeProcessingDdlHandlingPolicy": {
  "HandleSourceTableDropped": true,
  "HandleSourceTableTruncated": true,
  "HandleSourceTableAltered": true
},
```

Note

Untuk informasi tentang pernyataan DDL yang didukung untuk titik akhir tertentu, lihat topik yang menjelaskan titik akhir tersebut.

Pengaturan tugas substitusi karakter

Anda dapat menentukan bahwa tugas replikasi Anda melakukan substitusi karakter pada database target untuk semua kolom database sumber dengan AWS DMS STRING atau WSTRING tipe data. Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

Anda dapat mengkonfigurasi substitusi karakter untuk setiap tugas dengan endpoint dari sumber dan target database berikut:

- Database sumber:
 - Oracle
 - Microsoft SQL Server
 - MySQL
 - PostgreSQL
 - SAP Adaptive Server Enterprise (ASE)
 - IBM Db2 LUW
- Basisdata target:
 - Oracle
 - Microsoft SQL Server

- MySQL
- PostgreSQL
- SAP Adaptive Server Enterprise (ASE)
- Amazon Redshift

Anda dapat menentukan substitusi karakter menggunakan parameter `CharacterSetSettings` dalam pengaturan tugas Anda. Substitusi karakter ini terjadi untuk karakter yang ditentukan menggunakan nilai titik kode Unicode dalam notasi heksadesimal. Anda dapat menerapkan substitusi dalam dua tahap, dalam urutan berikut jika keduanya ditentukan:

1. Penggantian karakter individu - AWS DMS dapat mengganti nilai karakter yang dipilih pada sumber dengan nilai pengganti yang ditentukan dari karakter yang sesuai pada target. Gunakan susunan `CharacterReplacements` dalam `CharacterSetSettings` untuk memilih semua karakter sumber yang memiliki poin kode Unicode yang Anda tentukan. Gunakan susunan ini juga untuk menentukan poin kode pengganti untuk karakter yang sesuai pada target.

Untuk memilih semua karakter pada sumber yang memiliki titik kode tertentu, atur instans `SourceCharacterCodePoint` di susunan `CharacterReplacements` ke titik kode. Kemudian tentukan titik kode pengganti untuk semua karakter target setara dengan menetapkan instans yang sesuai dengan susunan `TargetCharacterCodePoint` ini. Untuk menghapus karakter target alih-alih menggantikannya, atur instans `TargetCharacterCodePoint` yang sesuai ke nol (0). Anda dapat mengganti atau menghapus banyak nilai yang berbeda dari karakter target yang Anda inginkan dengan menentukan pasangan pengaturan `SourceCharacterCodePoint` dan `TargetCharacterCodePoint` tambahan di susunan `CharacterReplacements`. Jika Anda menentukan nilai yang sama untuk beberapa instans `SourceCharacterCodePoint`, nilai dari pengaturan yang sesuai terakhir dari `TargetCharacterCodePoint` berlaku pada target.

Misalnya, anggaplah Anda menentukan nilai-nilai berikut untuk `CharacterReplacements`.

```
"CharacterSetSettings": {
  "CharacterReplacements": [ {
    "SourceCharacterCodePoint": 62,
    "TargetCharacterCodePoint": 61
  }, {
    "SourceCharacterCodePoint": 42,
    "TargetCharacterCodePoint": 41
  }
]
```

```
}

```

Dalam contoh ini, AWS DMS ganti semua karakter dengan nilai hex titik kode sumber 62 pada target dengan karakter dengan nilai titik kode 61. Juga, AWS DMS mengganti semua karakter dengan kode sumber titik 42 pada target dengan karakter dengan nilai titik kode 41. Dengan kata lain, AWS DMS menggantikan semua instans huruf 'b' pada target dengan huruf 'a'. Demikian pula, AWS DMS menggantikan semua contoh surat 'B' pada target dengan surat itu. 'A'

2. Validasi dan penggantian set karakter - Setelah penggantian karakter individu selesai, AWS DMS dapat memastikan bahwa semua karakter target memiliki poin kode Unicode yang valid dalam set karakter tunggal yang Anda tentukan. Anda menggunakan `CharacterSetSupport` di `CharacterSetSettings` untuk mengonfigurasi verifikasi dan modifikasi karakter target ini. Untuk menentukan set karakter verifikasi, atur `CharacterSet` di `CharacterSetSupport` dengan nilai string set karakter ini. Kemungkinan nilai untuk `CharacterSet` adalah sebagai berikut: Anda dapat AWS DMS memodifikasi karakter target yang tidak valid dengan salah satu cara berikut:

- Tentukan pengganti tunggal Unicode kode titik untuk semua karakter target tidak valid, terlepas dari titik kode mereka saat ini. Untuk mengonfigurasi titik kode pengganti ini, atur `ReplaceWithCharacterCodePoint` di `CharacterSetSupport` ke nilai yang ditentukan.
- Konfigurasi penghapusan semua karakter target yang tidak valid dengan mengatur `ReplaceWithCharacterCodePoint` ke nol (0).

Misalnya, anggaplah Anda menentukan nilai-nilai berikut untuk `CharacterSetSupport`.

```
"CharacterSetSettings": {
  "CharacterSetSupport": {
    "CharacterSet": "UTF16_PlatformEndian",
    "ReplaceWithCharacterCodePoint": 0
  }
}
```

Dalam contoh ini, AWS DMS menghapus karakter apa pun yang ditemukan pada target yang tidak valid di set karakter. "UTF16_PlatformEndian" Jadi, setiap karakter yang ditentukan dengan nilai hex 2FB6 dihapus. Nilai ini tidak valid karena ini adalah 4-byte Unicode kode titik dan UTF16 set karakter menerima hanya karakter dengan 2-byte kode poin.

Note

Tugas replikasi melengkapi semua substitusi karakter tertentu sebelum memulai transformasi global atau tabel-tingkat yang Anda tentukan melalui pemetaan tabel. Untuk informasi lebih lanjut tentang pemetaan tabel, lihat [Menggunakan pemetaan tabel untuk menentukan pengaturan tugas](#).

Substitusi karakter tidak mendukung tipe data LOB. Ini termasuk tipe data apa pun yang DMS anggap sebagai tipe data LOB. Misalnya, Extended tipe data di Oracle dianggap sebagai LOB. Untuk informasi selengkapnya tentang tipe data sumber, lihat berikut. [Jenis data sumber untuk Oracle](#)

Nilai-nilai yang AWS DMS mendukung untuk CharacterSet muncul dalam tabel berikut.

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_P12 A-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,s waplfnl
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

Pengaturan tugas citra sebelumnya

Saat menulis pembaruan CDC ke target streaming data seperti Kinesis atau Apache Kafka, Anda dapat melihat nilai asli baris basis data sumber sebelum diubah dengan pembaruan. Untuk memungkinkan hal ini, AWS DMS mengisi gambar sebelum peristiwa pembaruan berdasarkan data yang disediakan oleh mesin database sumber. Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

Untuk melakukannya, Anda menggunakan `BeforeImageSettings` parameter, yang menambahkan atribut JSON baru ke setiap operasi pembaruan dengan nilai yang dikumpulkan dari sistem database sumber.

Pastikan untuk menerapkan `BeforeImageSettings` hanya untuk beban penuh ditambah tugas CDC atau tugas CDC saja. Beban penuh ditambah tugas CDC memigrasikan data yang ada dan mereplikasi perubahan yang sedang berlangsung. Hanya tugas CDC yang mereplikasi perubahan data saja.

Jangan terapkan `BeforeImageSettings` pada tugas-tugas yang beban penuh saja.

Opsi yang memungkinkan untuk `BeforeImageSettings` adalah sebagai berikut:

- `EnableBeforeImage`— Menghidupkan sebelum pencitraan saat diatur ke `true`. Default adalah `false`.
- `FieldName` – Menetapkan nama untuk atribut JSON baru. Saat `EnableBeforeImage` adalah `true`, `FieldName` diperlukan dan tidak dapat kosong.
- `ColumnFilter` – Menentukan kolom untuk ditambahkan dengan menggunakan pencitraan sebelumnya. Untuk menambahkan hanya kolom yang merupakan bagian dari kunci utama tabel, gunakan nilai default, `pk-only`. Untuk menambahkan kolom apa pun yang memiliki nilai citra sebelum, gunakan `all`. Perhatikan bahwa gambar sebelum tidak mendukung tipe data objek biner besar (LOB) seperti CLOB dan BLOB.

Berikut ini menunjukkan contoh penggunaan `BeforeImageSettings`.

```
"BeforeImageSettings": {
  "EnableBeforeImage": true,
  "FieldName": "before-image",
  "ColumnFilter": "pk-only"
}
```

Untuk informasi tentang pengaturan citra sebelumnya untuk Kinesis, termasuk pengaturan pemetaan tabel tambahan, lihat [Menggunakan citra sebelumnya untuk melihat nilai asli dari baris CDC untuk Kinesis data stream sebagai target](#).

Untuk informasi tentang pengaturan citra sebelumnya untuk Kafka, termasuk pengaturan pemetaan tabel tambahan, lihat [Menggunakan citra sebelumnya untuk melihat nilai-nilai asli dari baris CDC untuk Apache Kafka sebagai target](#).

Penanganan kesalahan pengaturan tugas

Anda dapat mengatur perilaku penanganan kesalahan tugas replikasi Anda menggunakan pengaturan berikut. Untuk informasi tentang cara menggunakan file konfigurasi tugas untuk mengatur setelan tugas, lihat [Contoh pengaturan tugas](#).

- **DataErrorPolicy**— Menentukan tindakan yang diambil AWS DMS ketika ada kesalahan yang terkait dengan pemrosesan data pada tingkat catatan. Beberapa contoh kesalahan pemrosesan data termasuk kesalahan konversi, kesalahan dalam transformasi, dan data yang buruk. Default adalah LOG_ERROR.
 - IGNORE_RECORD – Tugas berlanjut dan data untuk catatan tersebut diabaikan. Penghitung kesalahan untuk properti DataErrorEscalationCount bertambah. Jadi, jika Anda menetapkan batas pada kesalahan untuk tabel, kesalahan ini dihitung ke batas itu.
 - LOG_ERROR – Tugas berlanjut dan kesalahan ditulis ke log tugas.
 - SUSPEND_TABLE – Tugas berlanjut tetapi data dari tabel dengan catatan kesalahan dipindahkan ke status kesalahan dan data tidak direplikasi.
 - STOP_TASK – Tugas berhenti dan intervensi manual diperlukan.
- **DataTruncationErrorPolicy** – Menentukan tindakan yang diambil AWS DMS saat data terpotong. Default adalah LOG_ERROR.
 - IGNORE_RECORD – Tugas berlanjut dan data untuk catatan tersebut diabaikan. Penghitung kesalahan untuk properti DataErrorEscalationCount bertambah. Jadi, jika Anda menetapkan batas pada kesalahan untuk tabel, kesalahan ini dihitung ke batas itu.
 - LOG_ERROR – Tugas berlanjut dan kesalahan ditulis ke log tugas.
 - SUSPEND_TABLE – Tugas berlanjut tetapi data dari tabel dengan catatan kesalahan dipindahkan ke status kesalahan dan data tidak direplikasi.
 - STOP_TASK – Tugas berhenti dan intervensi manual diperlukan.
- **DataErrorEscalationPolicy** – Menentukan tindakan yang diambil AWS DMS ketika jumlah maksimum kesalahan (diatur dalam parameter DataErrorEscalationCount) tercapai. Default adalah SUSPEND_TABLE.
 - SUSPEND_TABLE – Tugas berlanjut tetapi data dari tabel dengan catatan kesalahan dipindahkan ke status kesalahan dan data tidak direplikasi.
 - STOP_TASK – Tugas berhenti dan intervensi manual diperlukan.
- **DataErrorEscalationCount** – Menetapkan jumlah maksimum kesalahan yang dapat terjadi pada data untuk catatan tertentu. Ketika nomor ini tercapai, data untuk tabel yang

berisi catatan kesalahan ditangani sesuai dengan kebijakan yang ditetapkan dalam `DataErrorEscalationPolicy`. Default-nya adalah 0.

- `EventErrorPolicy`— Menentukan tindakan yang diambil AWS DMS ketika terjadi kesalahan saat mengirim acara terkait tugas. Nilai yang mungkin adalah
 - `IGNORE`— Tugas berlanjut dan data apa pun yang terkait dengan peristiwa itu diabaikan.
 - `STOP_TASK` – Tugas berhenti dan intervensi manual diperlukan.
- `TableErrorPolicy` – Menentukan tindakan yang diambil AWS DMS ketika terjadi kesalahan saat memproses data atau metadata untuk tabel tertentu. Kesalahan ini hanya berlaku untuk data tabel umum dan bukan merupakan kesalahan yang berkaitan dengan catatan tertentu. Default adalah `SUSPEND_TABLE`.
 - `SUSPEND_TABLE` – Tugas berlanjut tetapi data dari tabel dengan catatan kesalahan dipindahkan ke status kesalahan dan data tidak direplikasi.
 - `STOP_TASK` – Tugas berhenti dan intervensi manual diperlukan.
- `TableErrorEscalationPolicy` – Menentukan tindakan yang diambil AWS DMS ketika jumlah maksimum kesalahan (diatur menggunakan parameter `TableErrorEscalationCount`). Pengaturan default dan hanya pengguna adalah `STOP_TASK`, di mana tugas dihentikan dan intervensi manual diperlukan.
- `TableErrorEscalationCount` – Jumlah maksimum kesalahan yang dapat terjadi pada data umum atau metadata untuk tabel tertentu. Ketika nomor ini tercapai, data untuk tabel ditangani sesuai dengan kebijakan yang ditetapkan dalam `TableErrorEscalationPolicy`. Defaultnya adalah 0.
- `RecoverableErrorCount` – Jumlah maksimum upaya yang dilakukan untuk memulai ulang tugas ketika terjadi kesalahan lingkungan. Setelah sistem mencoba untuk me-restart tugas jumlah yang ditentukan kali, tugas dihentikan dan intervensi manual diperlukan. Nilai defaultnya adalah -1, yang menginstruksikan AWS DMS untuk mencoba memulai ulang tugas tanpa batas waktu. Menetapkan nilai ini ke 0 untuk tidak pernah mencoba untuk memulai ulang tugas. Jika terjadi kesalahan fatal, AWS DMS berhenti mencoba memulai kembali tugas setelah enam upaya.
- `RecoverableErrorInterval`— Jumlah detik yang menunggu AWS DMS antara upaya untuk memulai kembali tugas. Default-nya adalah 5.
- `RecoverableErrorThrottling`— Saat diaktifkan, interval antara upaya untuk memulai ulang tugas ditingkatkan dalam rangkaian berdasarkan nilai `RecoverableErrorInterval`. Misalnya, jika `RecoverableErrorInterval` diatur ke 5 detik, maka percobaan lagi berikutnya akan terjadi setelah 10 detik, lalu 20, lalu 40 detik dan seterusnya. Nilai default-nya `true`.

- `RecoverableErrorThrottlingMax`— Jumlah maksimum detik yang AWS DMS menunggu antara upaya untuk memulai kembali tugas jika `RecoverableErrorThrottling` diaktifkan. Default adalah 1800.
- `RecoverableErrorStopRetryAfterThrottlingMax`— Ketika diatur ke `true`, berhenti memulai ulang tugas setelah jumlah maksimum detik yang AWS DMS menunggu antara upaya pemulihan tercapai, per. `RecoverableErrorThrottlingMax`
- `ApplyErrorDeletePolicy` – Menentukan tindakan apa yang diambil AWS DMS ketika ada konflik dengan operasi DELETE. Default adalah `IGNORE_RECORD`. Nilai yang mungkin adalah sebagai berikut:
 - `IGNORE_RECORD` – Tugas berlanjut dan data untuk catatan tersebut diabaikan. Penghitung kesalahan untuk properti `ApplyErrorEscalationCount` bertambah. Jadi, jika Anda menetapkan batas pada kesalahan untuk tabel, kesalahan ini dihitung ke batas itu.
 - `LOG_ERROR` – Tugas berlanjut dan kesalahan ditulis ke log tugas.
 - `SUSPEND_TABLE` – Tugas berlanjut tetapi data dari tabel dengan catatan kesalahan dipindahkan ke status kesalahan dan data tidak direplikasi.
 - `STOP_TASK` – Tugas berhenti dan intervensi manual diperlukan.
- `ApplyErrorInsertPolicy` – Menentukan tindakan apa yang diambil AWS DMS ketika ada konflik dengan operasi INSERT. Default adalah `LOG_ERROR`. Nilai yang mungkin adalah sebagai berikut:
 - `IGNORE_RECORD` – Tugas berlanjut dan data untuk catatan tersebut diabaikan. Penghitung kesalahan untuk properti `ApplyErrorEscalationCount` bertambah. Jadi, jika Anda menetapkan batas pada kesalahan untuk tabel, kesalahan ini dihitung ke batas itu.
 - `LOG_ERROR` – Tugas berlanjut dan kesalahan ditulis ke log tugas.
 - `SUSPEND_TABLE` – Tugas berlanjut tetapi data dari tabel dengan catatan kesalahan dipindahkan ke status kesalahan dan data tidak direplikasi.
 - `STOP_TASK` – Tugas berhenti dan intervensi manual diperlukan.
 - `INSERT_RECORD` – Jika ada catatan target yang ada dengan kunci primer yang sama seperti catatan sumber yang dimasukkan, catatan target diperbarui.
- `ApplyErrorUpdatePolicy`— Menentukan tindakan apa yang diambil AWS DMS ketika ada konflik data yang hilang dengan operasi UPDATE. Nilai default-nya `LOG_ERROR`. Nilai yang mungkin adalah sebagai berikut:

- `IGNORE_RECORD` – Tugas berlanjut dan data untuk catatan tersebut diabaikan. Penghitung kesalahan untuk properti `ApplyErrorEscalationCount` bertambah. Jadi, jika Anda menetapkan batas pada kesalahan untuk tabel, kesalahan ini dihitung ke batas itu.
- `LOG_ERROR` – Tugas berlanjut dan kesalahan ditulis ke log tugas.
- `SUSPEND_TABLE` – Tugas berlanjut tetapi data dari tabel dengan catatan kesalahan dipindahkan ke status kesalahan dan data tidak direplikasi.
- `STOP_TASK` – Tugas berhenti dan intervensi manual diperlukan.
- `UPDATE_RECORD`— Jika catatan target hilang, catatan target yang hilang dimasukkan ke dalam tabel target. AWS DMS sepenuhnya menonaktifkan dukungan kolom LOB untuk tugas tersebut. Memilih opsi ini memerlukan supplemental logging penuh untuk diaktifkan untuk semua kolom tabel sumber ketika Oracle adalah basis data sumber.
- `ApplyErrorEscalationPolicy`— Menentukan tindakan apa yang diambil AWS DMS ketika jumlah maksimum kesalahan (diatur menggunakan `ApplyErrorEscalationCount` parameter) tercapai. Defaultnya adalah `LOG_ERROR`:
 - `LOG_ERROR` – Tugas berlanjut dan kesalahan ditulis ke log tugas.
 - `SUSPEND_TABLE` – Tugas berlanjut tetapi data dari tabel dengan catatan kesalahan dipindahkan ke status kesalahan dan data tidak direplikasi.
 - `STOP_TASK` – Tugas berhenti dan intervensi manual diperlukan.
- `ApplyErrorEscalationCount` – Opsi ini menetapkan jumlah maksimum konflik BERLAKU yang dapat terjadi untuk tabel tertentu selama operasi proses perubahan. Ketika nomor ini tercapai, tabel data ditangani sesuai dengan kebijakan yang ditetapkan dalam parameter `ApplyErrorEscalationPolicy`. Defaultnya adalah 0.
- `ApplyErrorFailOnTruncationDdl` – Atur opsi ini ke `true` untuk menyebabkan tugas gagal ketika pemotongan dilakukan pada salah satu tabel yang dilacak selama CDC. Default adalah `false`.

Pendekatan ini tidak bekerja dengan PostgreSQL versi 11.x atau lebih rendah, atau titik akhir sumber lain yang tidak mereplikasi pemotongan tabel DDL.

- `FailOnNoTablesCaptured`— Atur opsi ini `true` untuk menyebabkan tugas gagal saat pemetaan tabel yang ditentukan untuk tugas tidak menemukan tabel saat tugas dimulai. Default adalah `false`.
- `FailOnTransactionConsistencyBreached` – Opsi ini berlaku untuk tugas-tugas yang menggunakan Oracle sebagai sumber dengan CDC. Default-nya adalah salah. Atur ke `true` untuk

menyebabkan tugas gagal ketika transaksi terbuka dalam jangka waktu lebih lama dari batas waktu tertentu dan dapat dijatuhkan.

Ketika tugas CDC dimulai dengan Oracle, AWS DMS menunggu waktu terbatas untuk transaksi terbuka tertua ditutup sebelum memulai CDC. Jika transaksi terbuka tertua tidak ditutup sampai batas waktu tercapai, maka dalam banyak kasus AWS DMS dimulai CDC, mengabaikan transaksi itu. Jika opsi ini diatur ke `true`, tugas gagal.

- `FullLoadIgnoreConflicts`— Setel opsi ini `true` agar AWS DMS mengabaikan kesalahan “nol baris yang terpengaruh” dan “duplikat” saat menerapkan peristiwa yang di-cache. Jika disetel `false`, AWS DMS laporkan semua kesalahan alih-alih mengabaikannya. Nilai default-nya `true`.

Perhatikan bahwa kesalahan pemuatan tabel di Redshift sebagai target dilaporkan dalam `STL_LOAD_ERRORS` Untuk informasi selengkapnya, lihat [STL_LOAD_ERRORS di Panduan Pengembang Database Amazon Redshift](#).

Menyimpan pengaturan tugas

Anda dapat menyimpan pengaturan tugas sebagai file JSON jika Anda ingin menggunakan kembali pengaturan untuk tugas lain. Anda dapat menemukan pengaturan tugas untuk disalin ke file JSON di bawah bagian Detail Ikhtisar tugas.

Note

Saat menggunakan kembali pengaturan tugas untuk tugas lain, hapus `CloudWatchLogStream` atribut `CloudWatchLogGroup` dan apa pun. Jika tidak, kesalahan berikut diberikan: PESAN KESALAHAN SISTEM: Pengaturan Tugas `CloudWatchLogGroup` atau `CloudWatchLogStream` tidak dapat diatur saat membuat.

Sebagai contoh, file JSON berikut berisi pengaturan disimpan untuk tugas.

```
{
  "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
}

```

Mengatur dukungan LOB untuk database sumber dalam suatu tugas AWS DMS

Objek biner besar (LOB) kadang-kadang sulit untuk bermigrasi antar sistem. AWS DMS menawarkan sejumlah pilihan untuk membantu penyetelan kolom LOB. Untuk melihat tipe data mana dan kapan dianggap LOB AWS DMS, lihat AWS DMS dokumentasinya.

Ketika Anda bermigrasi data dari satu basis data ke yang lain, Anda dapat mengambil kesempatan untuk memikirkan kembali bagaimana LOB Anda disimpan, terutama untuk migrasi heterogen. Jika Anda ingin melakukannya, ada tidak perlu untuk bermigrasi LOB data.

Jika Anda memutuskan untuk memasukkan LOB, Anda kemudian dapat memutuskan pengaturan LOB lainnya:

- Mode LOB menentukan bagaimana LOB ditangani:
 - Mode LOB penuh - Dalam mode LOB penuh AWS DMS memigrasikan semua LOB dari sumber ke target terlepas dari ukurannya. Dalam konfigurasi ini, tidak AWS DMS memiliki informasi tentang ukuran maksimum LOB yang diharapkan. Dengan demikian, LOB dimigrasi satu per satu, sepotong demi sepotong. Mode LOB penuh bisa sangat lambat.
 - Mode LOB terbatas - Dalam mode LOB terbatas, Anda mengatur ukuran LOB maksimum untuk diterima DMS. Itu memungkinkan DMS untuk mengalokasikan memori dan memuat data LOB secara massal. LOB yang melebihi ukuran LOB maksimum terpotong, dan peringatan dikeluarkan ke file log. Dalam mode LOB terbatas, Anda dapat memperoleh kinerja yang signifikan melalui mode LOB penuh. Kami menyarankan agar Anda menggunakan mode LOB terbatas bila memungkinkan. Nilai maksimum yang disarankan adalah 102400 KB (100 MB).

 Note

Menggunakan opsi Max LOB size (K) dengan nilai lebih besar dari 63KB berdampak pada kinerja beban penuh yang dikonfigurasi untuk berjalan dalam mode LOB terbatas. Selama beban penuh, DMS mengalokasikan memori dengan mengalikan nilai ukuran LOB Maks (k) dengan tingkat Komit, dan produk dikalikan dengan jumlah kolom LOB. Ketika DMS tidak dapat mengalokasikan memori itu terlebih dahulu, DMS mulai mengkonsumsi memori SWAP, dan itu memengaruhi kinerja beban penuh. Jadi, jika Anda mengalami masalah kinerja saat menggunakan mode LOB terbatas, pertimbangkan untuk mengurangi tingkat komit hingga Anda mencapai tingkat kinerja yang dapat diterima. Anda juga dapat mempertimbangkan untuk menggunakan mode LOB sebaris untuk titik akhir yang didukung setelah Anda memahami distribusi LOB untuk tabel.

Untuk memvalidasi ukuran LOB terbatas, Anda harus menyetel `ValidationPartialLobSize` ke nilai yang sama dengan `LobMaxSize (K)`.

- Mode LOB sebaris - Dalam mode LOB sebaris, Anda mengatur ukuran LOB maksimum yang ditransfer DMS sebaris. LOB lebih kecil dari ukuran yang ditentukan ditransfer inline. LOB lebih besar dari ukuran yang ditentukan direplikasi menggunakan modus LOB penuh. Anda dapat memilih opsi ini untuk mereplikasi LOB kecil dan besar ketika sebagian besar LOB kecil. DMS tidak mendukung mode LOB sebaris untuk titik akhir yang tidak mendukung mode LOB Penuh, seperti S3 dan Redshift.

Note

Dengan Oracle, LOB diperlakukan sebagai tipe data VARCHAR bila memungkinkan. Pendekatan ini berarti bahwa AWS DMS mengambilnya dari database dalam jumlah besar, yang secara signifikan lebih cepat daripada metode lain. Ukuran maksimum VARCHAR di Oracle adalah 32 K. Oleh karena itu, ukuran LOB terbatas kurang dari 32 K optimal ketika Oracle adalah database sumber Anda.

- Ketika tugas dikonfigurasi untuk berjalan dalam mode LOB terbatas, opsi Ukuran LOB maks (K) menetapkan ukuran maksimum LOB yang diterima AWS DMS . Setiap LOB yang lebih besar dari nilai ini dipotong ke nilai ini.
- Ketika tugas dikonfigurasi untuk menggunakan mode LOB penuh, AWS DMS mengambil LOB dalam potongan-potongan. Opsi Ukuran potongan LOB (K) menentukan ukuran masing-masing bagian. Saat mengatur opsi ini, beri perhatian khusus pada ukuran paket maksimum yang diizinkan oleh konfigurasi jaringan Anda. Jika ukuran potongan LOB melebihi ukuran paket maksimum yang diizinkan, Anda mungkin melihat kesalahan pemutusan sambungan. Nilai yang disarankan untuk LobChunkSize adalah 64 kilobyte. Meningkatkan nilai di LobChunkSize atas 64 kilobyte dapat menyebabkan kegagalan tugas.
- Ketika tugas dikonfigurasi untuk berjalan dalam modus LOB inline, pengaturan InlineLobMaxSize menentukan LOB DMS ditransfer secara inline.

Note

Anda dapat menggunakan LOB tipe data hanya dengan tabel dan pandangan yang mencakup kunci primer.

Untuk informasi tentang pengaturan tugas untuk menentukan opsi ini, lihat [Menargetkan pengaturan tugas metadata](#)

Membuat beberapa tugas

Dalam beberapa skenario migrasi, Anda mungkin harus membuat beberapa tugas migrasi. Tugas bekerja secara independen dan dapat berjalan bersamaan. Setiap tugas memiliki sendiri beban awal, CDC, dan log membaca proses. Tabel yang terkait melalui manipulasi data bahasa (DMM) harus menjadi bagian dari tugas yang sama.

Beberapa alasan untuk membuat beberapa tugas untuk migrasi adalah sebagai berikut:

- Tabel target untuk tugas-tugas berada pada database yang berbeda, seperti ketika Anda mengipasi keluar atau melanggar sistem ke beberapa sistem.
- Anda ingin istirahat migrasi meja besar menjadi beberapa tugas dengan menggunakan penyaringan.

Note

Karena setiap tugas memiliki proses penangkapan perubahan dan pembacaan log sendiri, perubahan tidak terkoordinasi di seluruh tugas. Oleh karena itu, ketika menggunakan beberapa tugas untuk melakukan migrasi, pastikan bahwa setiap transaksi sumber individu sepenuhnya terkandung dalam satu tugas. Anda dapat menggunakan beberapa tugas untuk melakukan migrasi jika tidak ada transaksi individual yang dibagi menjadi tugas yang berbeda.

Membuat tugas untuk replikasi yang sedang berlangsung menggunakan AWS DMS

Anda dapat membuat AWS DMS tugas yang menangkap perubahan yang sedang berlangsung dari penyimpanan data sumber. Anda dapat melakukan penangkapan ini saat Anda memigrasi data Anda. Anda juga dapat membuat tugas yang menangkap perubahan yang sedang berlangsung setelah menyelesaikan migrasi awal (beban penuh) ke penyimpanan data target yang didukung. Proses ini disebut replikasi atau change data capture (CDC). AWS DMS menggunakan proses ini ketika mereplikasi perubahan yang sedang berlangsung dari penyimpanan data sumber. Proses ini bekerja dengan mengumpulkan perubahan log basis data menggunakan API asli mesin basis data.

Note

Anda dapat memigrasi tampilan hanya menggunakan tugas beban penuh. Jika tugas Anda adalah tugas hanya CDC atau tugas penuh beban yang dimulai CDC setelah selesai, migrasi hanya mencakup tabel dari sumber. Menggunakan full-load-only tugas, Anda dapat memigrasikan tampilan atau kombinasi tabel dan tampilan. Untuk informasi selengkapnya, lihat [Menentukan pemilihan tabel dan transformasi aturan menggunakan JSON](#).

Setiap mesin sumber memiliki persyaratan konfigurasi khusus untuk mengekspos aliran perubahan ini ke akun pengguna tertentu. Kebanyakan mesin memerlukan beberapa konfigurasi tambahan untuk memungkinkan proses capture mengkonsumsi data perubahan dengan cara yang berarti, tanpa kehilangan data. Sebagai contoh, Oracle membutuhkan penambahan penebangan tambahan, dan MySQL membutuhkan baris-tingkat biner logging (bin logging).

Untuk membaca perubahan yang sedang berlangsung dari basis data sumber, AWS DMS menggunakan tindakan API khusus mesin untuk membaca perubahan dari log transaksi mesin sumber. Berikut ini adalah beberapa contoh bagaimana AWS DMS melakukan itu:

- Untuk Oracle, AWS DMS gunakan API Oracle atau LogMiner API pembaca biner (bfile API) untuk membaca perubahan yang sedang berlangsung. AWS DMS membaca perubahan yang sedang berlangsung dari log pengulangan online atau arsip berdasarkan nomor perubahan sistem (SCN).
- Untuk Microsoft SQL Server, AWS DMS menggunakan MS-Replication atau MS-CDC untuk menulis informasi ke log transaksi SQL Server. Itu kemudian menggunakan fungsi `fn_dblog()` atau `fn_dump_dblog()` di SQL Server untuk membaca perubahan dalam log transaksi berdasarkan log sequence number (LSN).
- Untuk MySQL, AWS DMS membaca perubahan dari log biner berbasis baris (binlog) dan memigrasi perubahan tersebut ke target.
- Untuk PostgreSQL, AWS DMS menyiapkan slot replikasi logis dan menggunakan plugin `test_decoding` untuk membaca perubahan dari sumber dan memigrasi mereka ke target.
- Untuk Amazon RDS sebagai sumber, kami sarankan memastikan bahwa backup diaktifkan untuk mengatur CDC. Kami juga merekomendasikan untuk memastikan bahwa basis data sumber dikonfigurasi untuk mempertahankan log perubahan untuk waktu yang cukup—24 jam biasanya cukup. Untuk pengaturan khusus untuk setiap titik akhir, lihat berikut ini:
 - Amazon RDS for [Mengkonfigurasi sumber AWS Oracle yang dikelola untuk AWS DMS](#) Oracle:.
 - Amazon RDS untuk MySQL dan Aurora MySQL: [Menggunakan database yang kompatibel dengan MySQL AWS-managed sebagai sumber untuk AWS DMS](#)
 - Amazon RDS untuk SQL Server: [Menyiapkan replikasi yang sedang berlangsung pada instance cloud SQL Server DB](#)
 - Amazon RDS untuk PostgreSQL dan Aurora PostgreSQL: PostgreSQL secara otomatis menyimpan WAL yang diperlukan.

Ada dua jenis tugas replikasi yang sedang berlangsung:

- Beban penuh ditambah CDC – Tugas memigrasi data yang ada dan kemudian memperbarui basis data target berdasarkan perubahan basis data sumber.
- Hanya CDC – Tugas memigrasi perubahan yang sedang berlangsung setelah Anda memiliki data pada basis data target Anda.

Melakukan replikasi mulai dari titik awal CDC

Anda dapat memulai tugas replikasi AWS DMS yang sedang berlangsung (hanya change data capture) dari beberapa titik. Sumber daya yang dimaksud meliputi:

- Dari waktu mulai CDC kustom – Anda dapat menggunakan AWS Management Console atau AWS CLI untuk memberikan AWS DMS dengan timestamp di mana Anda ingin replikasi untuk dimulai. AWS DMS kemudian memulai tugas replikasi yang sedang berlangsung dari waktu mulai CDC kustom ini. AWS DMS mengonversi timestamp yang diberikan (dalam UTC) ke titik awal asli, seperti LSN untuk SQL Server atau SCN untuk Oracle. AWS DMS menggunakan metode khusus mesin untuk menentukan di mana untuk memulai tugas migrasi berdasarkan aliran perubahan mesin sumber.

Note

Hanya dengan menyetel atribut `StartFromContext` koneksi ke stempel waktu yang diperlukan, Db2 sebagai sumber menawarkan waktu mulai CDC yang disesuaikan. PostgreSQL sebagai sumber tidak mendukung waktu mulai CDC kustom. Hal ini karena mesin basis data PostgreSQL tidak memiliki cara untuk memetakan timestamp untuk LSN atau SCN seperti yang dilakukan Oracle dan SQL Server.

- Dari titik awal CDC asli – Anda juga dapat memulai dari titik asli di log transaksi mesin sumber. Dalam beberapa kasus, Anda mungkin lebih memilih pendekatan ini karena timestamp dapat menunjukkan beberapa titik asli dalam log transaksi. AWS DMS mendukung fitur ini untuk titik akhir sumber berikut:
 - Server SQL
 - PostgreSQL
 - Oracle
 - MySQL
 - MariaDB

Saat tugas dibuat, AWS DMS tandai titik awal CDC, dan itu tidak dapat diubah. Untuk menggunakan titik awal CDC yang berbeda, buat tugas baru.

Menentukan titik awal CDC asli

Titik awal asli CDC adalah titik di log mesin basis data yang mendefinisikan waktu di mana Anda dapat mulai CDC. Sebagai contoh, misalkan dump data massal telah diterapkan ke target. Anda dapat mencari titik awal asli untuk tugas khusus replikasi yang sedang berlangsung. Untuk menghindari inkonsistensi data, hati-hati memilih titik awal untuk tugas replikasi saja. DMS menangkap transaksi yang dimulai setelah titik awal CDC yang dipilih.

Berikut ini adalah contoh bagaimana Anda dapat menemukan CDC titik awal asli dari mesin sumber yang didukung:

Server SQL

Dalam SQL Server, log sequence number (LSN) memiliki tiga bagian:

- Nomor urutan berkas log virtual (VLF)
- Mulai offset dari blok log
- Nomor slot

Contoh LSN adalah sebagai berikut: `00000014:00000061:0001`

Untuk mendapatkan titik awal untuk tugas migrasi SQL Server berdasarkan pengaturan cadangan log transaksi Anda, gunakan fungsi `fn_dblog()` atau `fn_dump_dblog()` dalam SQL Server.

Untuk menggunakan titik awal asli CDC dengan SQL Server, buat publikasi pada tabel apa pun yang berpartisipasi dalam replikasi yang sedang berlangsung. AWS DMS membuat publikasi secara otomatis saat Anda menggunakan CDC tanpa menggunakan titik awal asli CDC.

PostgreSQL

Anda dapat menggunakan pemeriksaan pemulihan CDC untuk basis data sumber PostgreSQL Anda. Nilai titik pemeriksaan ini dihasilkan pada berbagai titik sebagai tugas replikasi yang sedang berlangsung dan berjalan untuk basis data sumber Anda (tugas induk). Untuk informasi lebih lanjut tentang titik pemeriksaan secara umum, lihat [Menggunakan titik pemeriksaan sebagai titik awal CDC](#).

Untuk mengidentifikasi titik pemeriksaan untuk digunakan sebagai titik awal asli Anda, gunakan basis data Anda `pg_replication_slots` atau detail gambaran umum tugas induk Anda dari AWS Management Console

Untuk menemukan detail gambaran umum untuk tugas induk Anda di konsol

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.

Jika Anda masuk sebagai pengguna IAM, pastikan Anda memiliki izin yang tepat untuk mengakses AWS DMS. Untuk informasi lebih lanjut tentang izin yang diperlukan, lihat [Izin IAM diperlukan untuk menggunakan AWS DMS](#).

2. Pada panel navigasi, pilih Tugas migrasi basis data.
3. Pilih tugas induk Anda dari daftar di halaman Tugas migrasi basis data. Melakukan hal ini akan membuka halaman tugas induk Anda, menampilkan detail gambaran umum.
4. Cari nilai titik pemeriksaan di bawah Change data capture (CDC), Posisi awal change data capture (CDC), dan Titik pemeriksaan pemulihan change data capture (CDC).

Nilai muncul serupa dengan berikut ini.

```
checkpoint:V1#1#000004AF/B00000D0#0#0#*#0#0
```

Di sini, komponen 4AF/B00000D0 adalah apa yang Anda butuhkan untuk menentukan titik awal CDC asli ini. Atur parameter API DMS `CdcStartPosition` ke nilai ini ketika Anda membuat tugas CDC untuk memulai replikasi pada titik awal ini untuk sumber PostgreSQL Anda. Untuk informasi tentang penggunaan AWS CLI untuk membuat tugas CDC ini, lihat [Mengaktifkan CDC dengan instans DB PostgreSQL terkelola AWS dengan AWS DMS](#).

Oracle

System change number (SCN) adalah stempel waktu internal yang logis, yang digunakan oleh basis data Oracle. SCN memesan peristiwa yang terjadi dalam basis data, yang diperlukan untuk memenuhi properti ACID transaksi. Oracle database menggunakan SCN untuk menandai lokasi di mana semua perubahan telah ditulis ke disk sehingga tindakan pemulihan tidak berlaku sudah ditulis perubahan. Oracle juga menggunakan SCN untuk menandai titik di mana ada mengulang ada untuk satu set data sehingga pemulihan dapat berhenti.

Untuk mendapatkan SCN saat ini dalam basis data Oracle, jalankan perintah berikut.

```
SELECT CURRENT_SCN FROM V$DATABASE
```

Jika Anda menggunakan SCN atau stempel waktu untuk memulai tugas CDC, Anda kehilangan hasil transaksi terbuka dan gagal memigrasikan hasil ini. Transaksi terbuka adalah transaksi yang dimulai sebelum posisi awal tugas dan dilakukan setelah posisi awal tugas. Anda dapat mengidentifikasi SCN dan timestamp untuk memulai tugas CDC pada titik yang mencakup semua transaksi terbuka. Untuk informasi selengkapnya, lihat [Transaksi](#) di dokumentasi Oracle online. Dengan versi 3.5.1 dan yang lebih tinggi, AWS DMS mendukung transaksi terbuka untuk tugas khusus CDC menggunakan pengaturan `openTransactionWindow` titik akhir jika Anda menggunakan SCN atau Timestamp untuk memulai tugas.

Saat menggunakan `openTransactionWindow` pengaturan, Anda harus menyediakan jendela, dalam beberapa menit, untuk menangani transaksi terbuka. AWS DMS menggeser posisi penangkapan dan menemukan posisi baru untuk memulai pengambilan data. AWS DMS menggunakan posisi awal baru untuk memindai setiap transaksi terbuka dari redo Oracle yang diperlukan atau log redo yang diarsipkan.

MySQL

Sebelum rilis MySQL versi 5.6.3, log sequence number (LSN) untuk MySQL adalah integer yang tidak diberi tanda 4-byte. Dalam MySQL versi 5.6.3, ketika batas ukuran file log redo meningkat dari 4 GB ke 512 GB, LSN menjadi integer yang tidak diberi tanda 8-byte. Peningkatan mencerminkan bahwa byte tambahan yang diperlukan untuk menyimpan informasi ukuran tambahan. Aplikasi yang dibangun di atas MySQL 5.6.3 atau lebih tinggi yang menggunakan nilai LSN harus menggunakan variabel 64-bit daripada 32-bit untuk menyimpan dan membandingkan nilai LSN. Untuk informasi selengkapnya tentang MySQL LSN, lihat [Dokumentasi MySQL](#).

Untuk mendapatkan LSN saat ini dalam basis data MySQL, jalankan perintah berikut.

```
mysql> show master status;
```

Kueri mengembalikan nama file binlog, posisi, dan beberapa nilai lainnya. Titik awal CDC asli adalah kombinasi dari nama file binlogs dan posisi, misalnya `mysql-bin-change1og.000024:373`. Dalam contoh ini, `mysql-bin-change1og.000024` adalah nama file binlogs dan `373` adalah posisi di mana AWS DMS perlu mulai menangkap perubahan.

Menggunakan titik pemeriksaan sebagai titik awal CDC

Tugas replikasi yang sedang berlangsung memigrasi perubahan, dan AWS DMS meng-cache informasi titik pemeriksaan yang khusus untuk AWS DMS dari waktu ke waktu. Titik pemeriksaan yang diciptakan AWS DMS berisi informasi sehingga mesin replikasi tahu titik pemulihan untuk

aliran perubahan. Anda dapat menggunakan titik pemeriksaan untuk kembali di timeline perubahan dan memulihkan tugas migrasi yang gagal. Anda juga dapat menggunakan titik pemeriksaan untuk memulai tugas replikasi lain yang sedang berlangsung untuk target lain pada titik tertentu dalam suatu waktu.

Anda bisa mendapatkan informasi pos pemeriksaan dengan salah satu dari tiga cara berikut:

- Jalankan operasi API `DescribeReplicationTasks` dan lihat hasilnya. Anda dapat memfilter informasi berdasarkan tugas dan mencari pos pemeriksaan. Anda dapat mengambil pos pemeriksaan terbaru ketika tugas dalam keadaan berhenti atau gagal. Informasi ini hilang jika tugas dihapus.
- Melihat tabel metadata bernama `awsdms_txn_state` pada instans target. Anda dapat mengkueri tabel untuk mendapatkan informasi titik pemeriksaan. Untuk membuat tabel metadata, atur parameter `TaskRecoveryTableEnabled` untuk `Yes` saat Anda membuat tugas. Pengaturan ini menyebabkan AWS DMS untuk terus menulis informasi titik pemeriksaan ke tabel metadata target. Informasi ini akan hilang jika tugas dihapus.

Misalnya, hal berikut ini adalah sampel titik pemeriksaan dalam tabel metadata:

```
checkpoint:V1#34#00000132/0F000E48#0#0#*#0#121
```

- Dari panel navigasi, pilih Tugas migrasi database, dan pilih tugas induk Anda dari daftar yang muncul di halaman Tugas migrasi database. Halaman tugas induk Anda terbuka, menampilkan detail ikhtisar. Cari nilai titik pemeriksaan di bawah Change data capture (CDC), Posisi awal change data capture (CDC), dan Titik pemeriksaan pemulihan change data capture (CDC). Nilai pos pemeriksaan tampak mirip dengan yang berikut:

```
checkpoint:V1#1#000004AF/B00000D0#0#0#*#0#0
```

Menghentikan tugas pada titik waktu komit atau server

Dengan diperkenalkannya titik awal CDC asli, AWS DMS juga dapat menghentikan tugas di titik-titik berikut:

- Waktu komit pada sumber
- Waktu server pada instans replikasi

Anda dapat mengubah tugas dan mengatur waktu di UTC untuk berhenti sesuai kebutuhan. Tugas secara otomatis berhenti berdasarkan komit atau server waktu yang Anda tetapkan. Atau, jika Anda

mengetahui waktu yang tepat untuk menghentikan tugas migrasi pada pembuatan tugas, Anda dapat mengatur waktu berhenti saat membuat tugas.

Note

Diperlukan waktu hingga 40 menit untuk menginisialisasi semua sumber daya saat pertama kali Anda memulai replikasi Tanpa AWS DMS Server baru. Perhatikan bahwa `server_time` opsi ini hanya berlaku setelah inisialisasi sumber daya selesai.

Melakukan replikasi dua arah

Anda dapat menggunakan tugas AWS DMS untuk melakukan replikasi dua arah antara dua sistem. Dalam replikasi dua arah, Anda mereplikasi data dari tabel yang sama (atau set tabel) antara dua sistem di kedua arah.

Sebagai contoh, Anda dapat menyalin tabel EMPLOYEE dari basis data A ke basis data B dan mereplikasi perubahan ke tabel dari basis data A ke basis data B. Anda juga dapat mereplikasi perubahan ke tabel EMPLOYEE dari basis data B kembali ke A. Dengan demikian, Anda melakukan replikasi dua arah.

Note

Replikasi AWS DMS dua arah tidak dimaksudkan sebagai solusi multi-master penuh termasuk simpul primer, resolusi konflik, dan sebagainya.

Menggunakan replikasi dua arah untuk situasi di mana data pada node yang berbeda dipisahkan secara operasional. Dengan kata lain, misalkan Anda memiliki elemen data diubah oleh aplikasi yang beroperasi pada node A, dan bahwa node A melakukan replikasi dua arah dengan node B. elemen data pada node A tidak pernah diubah oleh aplikasi yang beroperasi pada node B.

AWS DMS mendukung replikasi dua arah pada mesin database ini:

- Oracle
- SQL Server
- MySQL
- PostgreSQL

- Edisi yang kompatibel dengan Amazon Aurora MySQL
- yang kompatibel dengan Aurora PostgreSQL

Membuat tugas replikasi dua arah

Untuk mengaktifkan AWS DMS dua arah replikasi, mengkonfigurasi sumber dan target akhir untuk kedua database (A dan B). Sebagai contoh, mengkonfigurasi titik akhir sumber untuk database A, titik akhir sumber untuk database B, titik akhir target untuk database A, dan titik akhir target untuk database B.

Kemudian membuat dua tugas: satu tugas untuk sumber A untuk memindahkan data ke target B, dan tugas lain untuk sumber B untuk memindahkan data ke target A. juga, pastikan bahwa setiap tugas dikonfigurasi dengan pencegahan loopback. Melakukan hal ini mencegah perubahan identik dari yang diterapkan ke target dari kedua tugas, sehingga merusak data untuk setidaknya satu dari mereka. Untuk informasi selengkapnya, lihat [Mencegah loopback](#).

Untuk pendekatan termudah, mulai dengan dataset identik pada kedua database A dan database B. kemudian membuat dua CDC hanya tugas, satu tugas untuk mereplikasi data dari A ke B, dan tugas lain untuk mereplikasi data dari B ke A.

Untuk menggunakan AWS DMS untuk instantiate dataset baru (database) pada node B dari node A, lakukan hal berikut:

1. Menggunakan beban penuh dan CDC tugas untuk memindahkan data dari database A ke B. Pastikan bahwa tidak ada aplikasi memodifikasi data pada database B selama waktu ini.
2. Ketika beban penuh selesai dan sebelum aplikasi diperbolehkan untuk memodifikasi data pada database B, perhatikan waktu atau CDC mulai posisi database B. untuk petunjuk, lihat [Melakukan replikasi mulai dari titik awal CDC](#).
3. Membuat CDC hanya tugas yang bergerak data dari database B kembali ke A menggunakan ini mulai waktu atau CDC mulai posisi.

Note

Hanya satu tugas dalam pasangan dua arah dapat beban penuh dan CDC.

Mencegah loopback

Untuk menunjukkan mencegah loopback, anggaplah bahwa dalam T1 tugas AWS DMS membaca log perubahan dari sumber database A dan menerapkan perubahan untuk menargetkan database B.

Berikutnya, tugas kedua, T2, membaca perubahan log dari sumber database B dan berlaku mereka kembali ke target database A. sebelum T2 melakukan ini, DMS harus memastikan bahwa perubahan yang sama dibuat untuk menargetkan database B dari sumber database A tidak dibuat untuk sumber database A. dengan kata lain, DMS harus memastikan bahwa perubahan ini aren tidak bergema (dilingkarkan) kembali ke basis data target A. jika tidak, data dalam database A dapat rusak.

Untuk mencegah loopback perubahan, tambahkan pengaturan tugas berikut untuk setiap tugas replikasi dua arah. Melakukan hal ini memastikan bahwa loopback data korupsi tidak terjadi di kedua arah.

```
{
  . . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": Boolean,
    "SourceSchema": String,
    "TargetSchema": String
  },

  . . .
}
```

Parameter `LoopbackPreventionSettings` menentukan jika transaksi baru atau gema dari tugas replikasi berlawanan. Saat AWS DMS menerapkan transaksi ke database target, itu update tabel DMS (`awsdms_loopback_prevention`) dengan indikasi perubahan. Sebelum menerapkan setiap transaksi ke target, DMS mengabaikan setiap transaksi yang mencakup referensi ke `awsdms_loopback_prevention` tabel. Oleh karena itu, tidak berlaku perubahan.

Termasuk pengaturan tugas ini dengan setiap replikasi tugas dalam pasangan dua arah. Pengaturan ini memungkinkan pencegahan loopback. Mereka juga menentukan skema untuk setiap sumber dan target database dalam tugas yang mencakup `awsdms_loopback_prevention` tabel untuk setiap titik akhir.

Untuk membolehkan setiap tugas untuk mengenal pasti gema itu dan membuangnya, tetapkan `EnableLoopbackPrevention` ke `true`. Untuk menentukan skema pada sumber yang

mencakup `aws_dms_loopback_prevention`, atur `SourceSchema` untuk nama untuk skema tersebut dalam database sumber. Untuk menentukan skema pada target yang mencakup tabel yang sama, atur `TargetSchema` untuk skema tersebut dalam database target.

Dalam contoh berikut, `SourceSchema` dan `TargetSchema` pengaturan untuk replikasi tugas T1 dan replikasi berlawanan T2 tugas ditentukan dengan pengaturan berlawanan.

Tetapan untuk tugas T1 adalah seperti berikut.

```
{
  . . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": true,
    "SourceSchema": "LOOP-DATA",
    "TargetSchema": "loop-data"
  },

  . . .
}
```

Pengaturan untuk tugas yang berlawanan T2 adalah sebagai berikut.

```
{
  . . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": true,
    "SourceSchema": "loop-data",
    "TargetSchema": "LOOP-DATA"
  },

  . . .
}
```

Note

Ketika menggunakan AWS CLI, gunakan hanya `create-replication-task` atau `modify-replication-task` perintah untuk mengkonfigurasi `LoopbackPreventionSettings` dalam tugas replikasi dua arah Anda.

Batasan replikasi dua arah

Replikasi dua arah untuk AWS DMS memiliki batasan sebagai berikut:

- Pencegahan loopback hanya melacak data language manipulation language (DML) pernyataan. AWS DMS tidak mendukung pencegah data definition language (DDL) loopback. Untuk melakukannya, konfigurasi salah satu tugas dalam pasangan dua arah untuk memfilter pernyataan DDL.
- Tugas yang menggunakan pencegahan loopback tidak mendukung perubahan dalam batch. Untuk mengkonfigurasi tugas dengan pencegahan loopback, pastikan untuk mengatur `BatchApplyEnabled` ke `false`.
- DMS dua arah replikasi tidak termasuk konflik deteksi atau resolusi. Untuk mendeteksi inkonsistensi data, menggunakan validasi data pada kedua tugas.

Mengubah tugas

Anda dapat mengubah tugas jika Anda perlu mengubah pengaturan tugas, pemetaan tabel, atau pengaturan lainnya. Anda juga dapat mengaktifkan dan menjalankan penilaian perdana sebelum menjalankan tugas yang dimodifikasi. Anda dapat memodifikasi tugas di konsol dengan memilih tugas dan memilih `Memodifikasi`. Anda juga dapat menggunakan perintah CLI atau operasi API.

[ModifyReplicationTask](#)

Ada beberapa keterbatasan untuk memodifikasi tugas. Sumber daya yang dimaksud meliputi:

- Anda tidak dapat memodifikasi sumber atau target endpoint tugas.
- Anda tidak dapat mengubah jenis tugas migrasi.
- Tugas yang telah dijalankan harus memiliki status `dihentikan` atau `Gagal` untuk dimodifikasi.

Memindahkan tugas

Anda dapat memindahkan tugas ke contoh replikasi yang berbeda ketika salah satu situasi berikut berlaku untuk kasus penggunaan Anda.

- Anda sedang menggunakan sebuah instance dari jenis tertentu dan Anda ingin beralih ke jenis instans yang berbeda.

- Contoh Anda saat ini kelebihan beban oleh banyak replikasi tugas, dan Anda ingin membagi beban di beberapa contoh.
- Penyimpanan instans Anda penuh, dan Anda ingin memindahkan tugas dari instance tersebut ke contoh yang lebih kuat sebagai alternatif untuk penskalaan penyimpanan atau komputasi.
- Anda ingin menggunakan fitur yang baru dirilis dari AWS DMS, namun tidak ingin membuat tugas baru dan memulai ulang migrasi. Sebaliknya, Anda lebih memilih untuk spin up contoh replikasi dengan baru AWS DMS versi yang mendukung fitur, dan memindahkan tugas yang ada ke contoh itu.

Anda dapat memindahkan tugas di konsol dengan memilih tugas dan memilih **Pindahkan**. Anda juga dapat menjalankan perintah CLI atau operasi API `MoveReplicationTask` untuk memindahkan tugas. Anda dapat memindahkan tugas yang memiliki mesin database sebagai titik akhir targetnya.

Pastikan bahwa contoh replikasi target memiliki cukup ruang penyimpanan untuk mengakomodasi tugas yang sedang dipindahkan. Jika tidak, skala penyimpanan untuk membuat ruang untuk contoh replikasi target Anda sebelum memindahkan tugas.

Juga, pastikan bahwa instance replikasi target Anda dibuat dengan versi AWS DMS engine yang sama atau lebih tinggi dengan instance replikasi saat ini.

Note

- Anda tidak dapat memindahkan tugas ke contoh replikasi yang sama di mana saat ini berada.
- Anda tidak dapat memodifikasi pengaturan tugas saat sedang dipindahkan.
- Tugas yang telah Anda jalankan harus memiliki status **dihentikan**, **Gagal**, atau **Pemindahan Gagal** sebelum Anda dapat memindahkannya.

Ada dua status tugas yang berhubungan dengan memindahkan tugas DMS, **MemindahkandanPemindahan Gagal**. Untuk informasi selengkapnya tentang status pekerjaan, lihat [Status Tugas](#).

Setelah memindahkan tugas, Anda dapat mengaktifkan dan menjalankan penilaian perdana untuk memeriksa masalah pemblokiran sebelum menjalankan tugas yang dipindahkan.

Memuat ulang tabel selama mengerjakan tugas

Sementara tugas berjalan, Anda dapat reload tabel database target menggunakan data dari sumber. Anda mungkin ingin memuat ulang tabel jika, selama tugas, terjadi kesalahan atau perubahan data karena operasi partisi (misalnya, ketika menggunakan Oracle). Anda dapat memuat ulang hingga 10 tabel dari tugas.

Memuat ulang tabel tidak menghentikan tugas.

Untuk memuat ulang tabel, ketentuan berikut harus berlaku:

- Tugas harus berjalan.
- Metode migrasi untuk tugas harus baik beban penuh atau beban penuh dengan CDC.
- Tabel duplikat tidak diizinkan.
- AWS DMS mempertahankan definisi tabel dibaca sebelumnya dan tidak menciptakan itu selama operasi reload. Setiap pernyataan DDL seperti ALTER TABLE ADD KOLOM atau DROP KOLOM yang dibuat ke meja sebelum meja reloaded dapat menyebabkan operasi reload gagal.

Note

DMS menerapkan `TargetTablePrepMode` pengaturan sebelum memuat ulang tabel. Jika Anda mengatur `TargetTablePrepMode` ke `DO_NOTHING`, Anda harus memotong tabel secara manual terlebih dahulu.

AWS Management Console

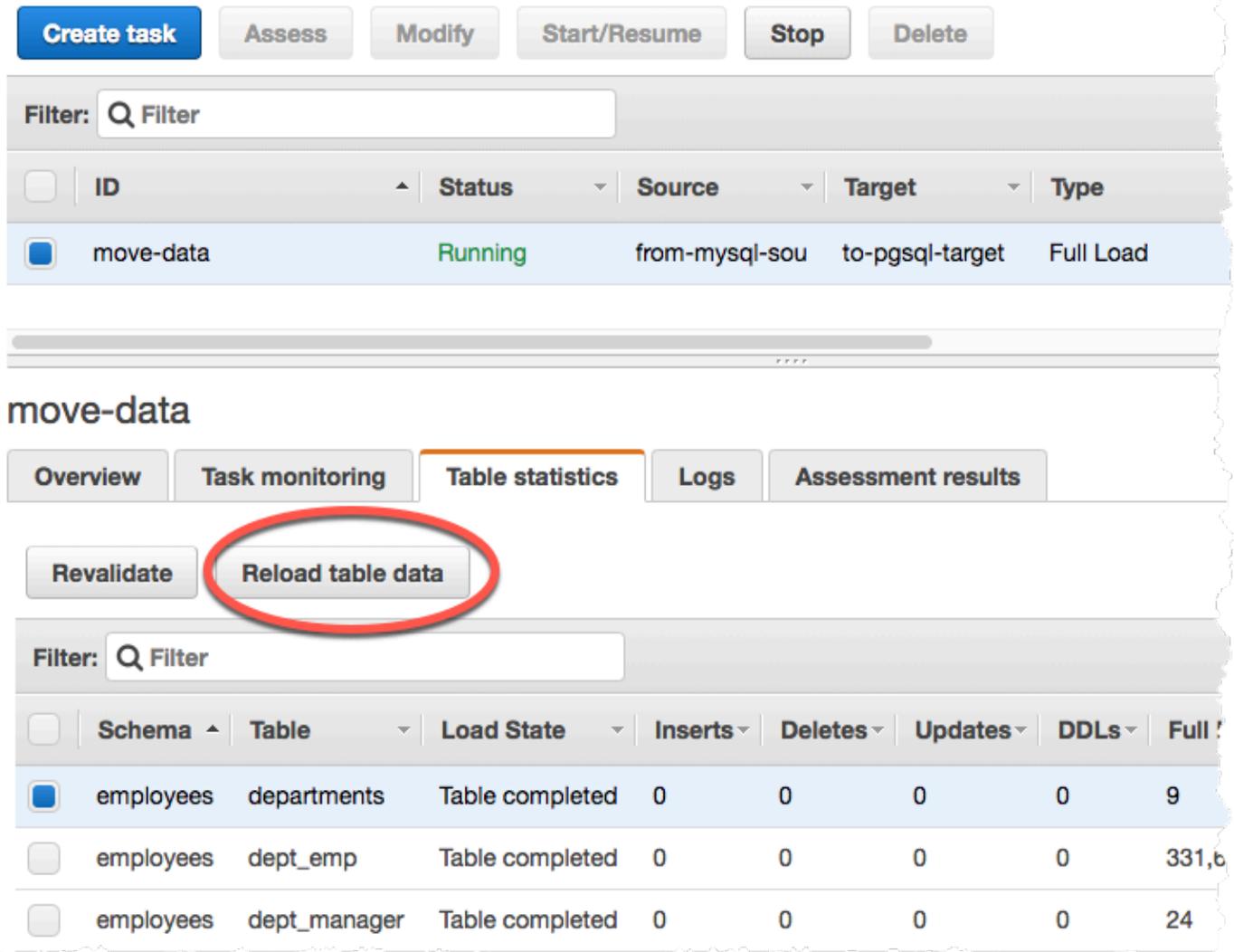
Untuk reload tabel menggunakan AWS DMS Konsol

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.

Jika Anda masuk sebagai pengguna IAM, pastikan Anda memiliki izin yang tepat untuk mengakses AWS DMS. Untuk informasi lebih lanjut tentang izin yang diperlukan, lihat [Izin IAM diperlukan untuk menggunakan AWS DMS](#).

2. Di panel navigasi, pilih Pengguna.
3. Pilih tugas berjalan yang memiliki tabel yang ingin Anda reload.

4. PilihTabel Statistiktab.



The screenshot shows the AWS DMS console interface. At the top, there are buttons for 'Create task', 'Assess', 'Modify', 'Start/Resume', 'Stop', and 'Delete'. Below these is a search filter. A table lists tasks, with 'move-data' selected. Below the task list, the 'move-data' task details are shown, including tabs for 'Overview', 'Task monitoring', 'Table statistics', 'Logs', and 'Assessment results'. The 'Table statistics' tab is active, and the 'Reload table data' button is circled in red. Below this is another search filter and a table of table statistics.

ID	Status	Source	Target	Type
move-data	Running	from-mysql-sou	to-pgsq-target	Full Load

move-data

Overview Task monitoring **Table statistics** Logs Assessment results

Revalidate **Reload table data**

Schema	Table	Load State	Inserts	Deletes	Updates	DDLs	Full
employees	departments	Table completed	0	0	0	0	9
employees	dept_emp	Table completed	0	0	0	0	331,6
employees	dept_manager	Table completed	0	0	0	0	24

5. Pilih tabel yang ingin Anda reload. Jika tugas tidak lagi berjalan, Anda tidak dapat memuat ulang tabel.

6. Pilih Muat ulang data tabel.

Saat AWS DMS sedang mempersiapkan untuk reload tabel, konsol mengubah status tabel untuk Tabel sedang dimuat ulang.

Menggunakan pemetaan tabel untuk menentukan pengaturan tugas

Tabel pemetaan menggunakan beberapa jenis aturan untuk menentukan sumber data, skema sumber, data, dan setiap transformasi yang harus terjadi selama tugas. Anda dapat menggunakan pemetaan tabel untuk menentukan tabel individu dalam database untuk bermigrasi dan skema yang digunakan untuk migrasi.

Ketika bekerja dengan pemetaan tabel, Anda dapat menggunakan filter untuk menentukan data yang ingin direplikasi dari kolom tabel. Selain itu, Anda dapat menggunakan transformasi untuk memodifikasi dipilih skema, tabel, atau tampilan sebelum mereka ditulis ke database target.

Topik

- [Menentukan pemilihan tabel dan transformasi aturan dari konsol](#)
- [Menentukan pemilihan tabel dan transformasi aturan menggunakan JSON](#)
- [Aturan seleksi dan tindakan](#)
- [Wildcard dalam pemetaan tabel](#)
- [Aturan dan tindakan transformasi](#)
- [Menggunakan ekspresi aturan transformasi untuk menentukan isi kolom](#)
- [Tabel dan koleksi pengaturan aturan dan operasi](#)

Note

Ketika bekerja dengan pemetaan meja untuk endpoint sumber MongoDB, Anda dapat menggunakan filter untuk menentukan data yang ingin direplikasi, dan menentukan nama database di tempatschema_name. Atau, Anda dapat menggunakan default"%".

Menentukan pemilihan tabel dan transformasi aturan dari konsol

Anda dapat menggunakan AWS Management Console untuk melakukan pemetaan tabel, termasuk menentukan pemilihan tabel dan transformasi. Pada konsol, gunakan `Di manabagian` untuk menentukan skema, tabel, dan tindakan (termasuk atau mengecualikan). Gunakan `Filterbagian` untuk menentukan nama kolom dalam tabel dan kondisi yang ingin Anda terapkan untuk tugas replikasi. Bersama-sama, kedua tindakan ini membuat aturan seleksi.

Anda dapat menyertakan transformasi dalam pemetaan tabel setelah Anda telah ditentukan setidaknya satu aturan pilihan. Anda dapat menggunakan transformasi untuk mengubah nama skema atau tabel, menambahkan awalan atau akhiran untuk skema atau tabel, atau menghapus kolom tabel.

 Note

AWS DMS tidak mendukung lebih dari satu aturan transformasi per tingkat skema, tingkat tabel, atau tingkat kolom.

Prosedur berikut menunjukkan cara mengatur aturan seleksi, berdasarkan tabel yang disebut **Customers** dalam skema yang disebut **EntertainmentAgencySample**.

Untuk menentukan pilihan tabel, kriteria filter, dan transformasi menggunakan konsol

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.

Jika Anda masuk sebagai pengguna IAM, pastikan Anda memiliki izin yang tepat untuk mengakses AWS DMS. Untuk informasi lebih lanjut tentang izin yang diperlukan, lihat [Izin IAM diperlukan untuk menggunakan AWS DMS](#).

2. Pada halaman Dasbor, pilih tugas migrasi database.
3. Pilih **Buat** tugas.
4. Di bagian Konfigurasi tugas, masukkan informasi tugas, termasuk pengidentifikasi tugas, contoh replikasi, titik akhir basis data sumber, titik akhir basis data target, dan tipe Migrasi.

DMS > Database migration tasks > Create database migration task

Create database migration task

Task configuration

Task identifier

Type a unique identifier for the task

Replication instance

Choose a replication instance

Source database endpoint

Choose a source database endpoint

Target database endpoint

Choose a target database endpoint

Migration type [Info](#)

Migrate existing data

5. Di bagian Pemetaan tabel, masukkan nama skema dan nama tabel. Anda dapat menggunakan “%” sebagai nilai wildcard saat menentukan nama skema atau nama tabel. Untuk informasi tentang wildcard lain yang dapat Anda gunakan, lihat [the section called “Wildcard dalam pemetaan tabel”](#). Tentukan tindakan yang akan diambil, untuk menyertakan atau mengecualikan data yang didefinisikan oleh filter.

Table mappings

Editing mode [Info](#)

Wizard
You can enter only a subset of the available table mappings.

JSON editor
You can enter all available table mappings directly in JSON format.

Specify at least one selection rule with an include action. After you do this, you can add one or more transformation rules.

▼ Selection rules

Choose the schema and/or tables you want to include with, or exclude from, your migration task. [Info](#)

Add new selection rule

▼ where **schema name** is like 'MySchema' and **table name** is like '%', include

Schema
Enter a schema

Schema name
Use the % character as a wildcard
MySchema

Table name
Use the % character as a wildcard
%

Action
Choose "Include" to migrate your selected objects, or "Exclude" to ignore them during the migration.
Include

6. Tentukan informasi filter menggunakan
 - a. Pilih Tambahkan filter kolom untuk menentukan kolom dan kondisi.
 - b. Pilih Tambahkan kondisi untuk menambahkan kondisi tambahan.

Contoh berikut menunjukkan filter untuk **Customer** tabel yang mencakup **Agency IDs** antara **01** dan **85**.

Source filters [Info](#) Add column filter

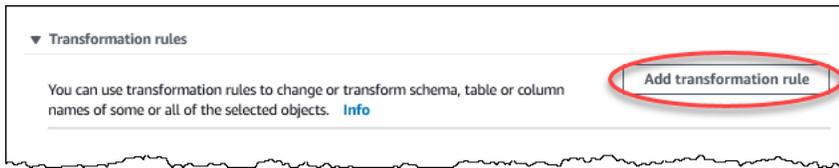
▼ Column filter 1 ×

Column name
AgencyId

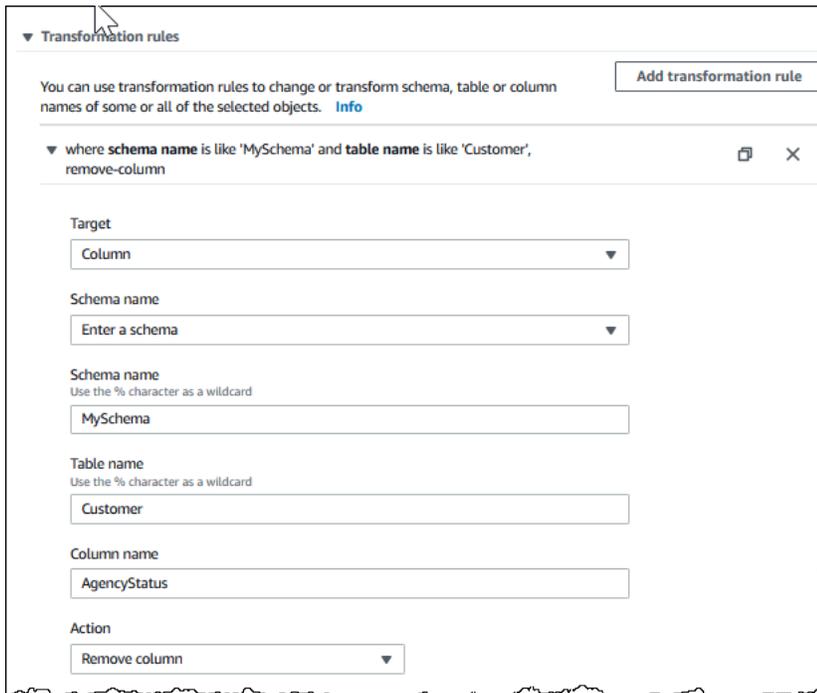
Condition 1
Equal to or between two values 01
85

Add condition

7. Ketika Anda telah membuat pilihan yang Anda inginkan, pilih Tambahkan aturan pilihan baru.
8. Setelah Anda telah membuat setidaknya satu aturan pilihan, Anda dapat menambahkan transformasi untuk tugas. Pilih tambahkan aturan transformasi.



- Pilih target yang ingin Anda ubah, dan masukkan informasi tambahan yang diminta. Contoh berikut menunjukkan transformasi yang menghapus **AgencyStatus** kolom dari **Customer** Tabel.



- Pilih Tambahkan aturan transformasi.

- Pilih Buat tugas.

Note

AWS DMS tidak mendukung lebih dari satu aturan transformasi per tingkat skema atau per tingkat tabel.

Menentukan pemilihan tabel dan transformasi aturan menggunakan JSON

Untuk menentukan pemetaan tabel yang ingin Anda terapkan selama migrasi, Anda dapat membuat file JSON. Jika Anda membuat tugas migrasi menggunakan konsol, Anda dapat menelusuri file JSON ini atau memasukkan JSON langsung ke kotak pemetaan tabel. Jika Anda menggunakan

CLI atau API untuk melakukan migrasi, Anda dapat menentukan file ini menggunakan parameter `TableMappings` dari operasi API `CreateReplicationTask` atau `ModifyReplicationTask`.

AWS DMS hanya dapat memproses pemetaan tabel file JSON hingga ukuran 2 MB. Kami menyarankan Anda menjaga aturan pemetaan ukuran file JSON di bawah batas 2 MB saat bekerja dengan tugas DMS. Ini mencegah kesalahan tak terduga selama pembuatan atau modifikasi tugas. Ketika file aturan pemetaan melebihi batas 2 MB, kami sarankan Anda membagi tabel di beberapa tugas untuk mengurangi ukuran file aturan pemetaan sehingga tetap di bawah batas ini.

Anda dapat menentukan apa tabel, pandangan, dan skema Anda ingin bekerja dengan. Anda juga dapat melakukan tabel, tampilan, dan skema transformasi dan menentukan pengaturan untuk bagaimana AWS DMS beban tabel individu dan pandangan. Anda membuat aturan pemetaan tabel untuk opsi ini menggunakan jenis aturan berikut:

- `selectionrules` - Mengidentifikasi jenis dan nama tabel sumber, pandangan, dan skema untuk memuat. Untuk informasi selengkapnya, lihat [Aturan seleksi dan tindakan](#).
- `transformationrules` - Tentukan perubahan tertentu atau penambahan tabel sumber tertentu dan skema pada sumber sebelum mereka dimuat pada target. Untuk informasi selengkapnya, lihat [Aturan dan tindakan transformasi](#).

Juga, untuk menentukan isi kolom baru dan yang sudah ada, Anda dapat menggunakan ekspresi dalam aturan transformasi. Untuk informasi selengkapnya, lihat [Menggunakan ekspresi aturan transformasi untuk menentukan isi kolom](#).

- `table-settingsrules` - Tentukan bagaimana tugas DMS memuat data untuk tabel individu. Untuk informasi selengkapnya, lihat [Tabel dan koleksi pengaturan aturan dan operasi](#).

Note

Untuk target Amazon S3, Anda juga dapat menandai objek S3 dipetakan ke tabel yang dipilih dan skema menggunakan `post-processing` aturan jenis `add-tag` Tindakan aturan.

Untuk informasi selengkapnya, lihat [Penandaan objek Amazon S3](#).

Untuk target berikut, Anda dapat menentukan bagaimana dan di mana dipilih skema dan tabel dimigrasi ke target menggunakan `object-mapping` Jenis aturan:

- Amazon DynamoDB — Untuk informasi lebih lanjut, lihat [Menggunakan pemetaan objek untuk migrasi data ke DynamoDB](#).

- Amazon Kinesis — Untuk informasi lebih lanjut, lihat [Menggunakan pemetaan objek untuk memigrasi data ke Kinesis data stream](#).
- Apache Kafka - Untuk informasi lebih lanjut, lihat [Menggunakan pemetaan objek untuk bermigrasi data ke topik Kafka](#).

Aturan seleksi dan tindakan

Menggunakan pemetaan tabel, Anda dapat menentukan apa tabel, pandangan, dan skema Anda ingin bekerja dengan dengan menggunakan aturan seleksi dan tindakan. Untuk aturan pemetaan tabel yang menggunakan jenis aturan seleksi, Anda dapat menerapkan nilai berikut.

Parameter	Kemungkinan nilai	Deskripsi
<code>rule-type</code>	<code>selection</code>	Aturan seleksi. Tentukan setidaknya satu aturan pilihan ketika menentukan pemetaan tabel.
<code>rule-id</code>	Nilai numerik.	Nilai numerik yang unik untuk mengidentifikasi aturan.
<code>rule-name</code>	Nilai alfanumerik.	Nama unik untuk mengidentifikasi aturan.
<code>rule-action</code>	<code>include, exclude, explicit</code>	Nilai yang mencakup atau mengecualikan objek atau objek yang dipilih oleh aturan. Jika <code>explicit</code> ditentukan, Anda dapat memilih dan mencakup hanya satu objek yang sesuai dengan tabel eksplisit ditentukan dan skema.
<code>object-locator</code>	Objek dengan parameter berikut: <ul style="list-style-type: none"> • <code>schema-name</code> — Nama skema. • <code>table-name</code> — Nama tabel. • (Opsional) <code>table-type</code> — <code>table</code> <code>view</code> <code>all</code>, untuk menunjukk 	Nama setiap skema dan tabel atau tampilan yang aturan berlaku. Anda juga dapat menentukan apakah aturan hanya mencakup tabel, hanya pandangan, atau kedua tabel dan pandangan. Jika <code>rule-action</code>

Parameter	Kemungkinan nilai	Deskripsi
	<p>an jika <code>table-name</code> hanya mengacu pada tabel, pandangan, atau kedua tabel dan pandangan. Default adalah <code>table</code>.</p> <p>AWS DMS memuat tampilan hanya dalam tugas beban penuh. Jika Anda hanya memiliki tugas full load dan change data capture (CDC), konfigurasi setidaknya satu full-load-only tugas untuk memuat tampilan Anda.</p> <p>Tidak semua titik akhir target menerima tampilan sebagai sumber replikasi, bahkan dalam beban penuh (misalnya OpenSearch Layanan Amazon). Periksa batasan titik akhir target Anda.</p>	<p>adalah <code>include</code> atau <code>exclude</code>, Anda dapat menggunakan tanda persen “%” sebagai wildcard untuk semua atau sebagian dari nilai untuk parameter <code>schema-name</code> dan <code>table-name</code>. Untuk informasi tentang wildcard lain yang dapat Anda gunakan, lihat the section called “Wildcard dalam pemetaan tabel”. Dengan demikian, Anda bisa mencocokkan item berikut:</p> <ul style="list-style-type: none"> • Sebuah tabel tunggal, tampilan, atau koleksi dalam skema tunggal • Sebuah tabel tunggal, tampilan, atau koleksi dalam beberapa atau semua skema • Beberapa atau semua tabel dan tampilan dalam skema tunggal, atau koleksi dalam satu database • Beberapa atau semua tabel dan tampilan dalam beberapa atau semua skema, atau koleksi di beberapa atau semua database <p>Jika <code>rule-action</code> adalah <code>explicit</code>, Anda hanya dapat menentukan nama yang tepat dari tabel tunggal atau tampilan dan skema (tanpa wildcard).</p> <p>Sumber yang didukung untuk tampilan meliputi:</p> <ul style="list-style-type: none"> • Oracle

Parameter	Kemungkinan nilai	Deskripsi
		<ul style="list-style-type: none"> • Microsoft SQL Server • PostgreSQL • IBM Db2 LUW • IBM Db2 z/OS • SAP Adaptive Server Enterprise (ASE) • MySQL • AURORA • AURORA Tanpa Server • MariaDB <div data-bbox="976 827 1508 1329" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>AWS DMS tidak pernah memuat tampilan sumber ke tampilan target. Sebuah tampilan sumber dimuat ke tabel setara pada target dengan nama yang sama dengan tampilan pada sumber.</p> </div> <p>Sumber yang didukung untuk database yang berisi koleksi meliputi:</p> <ul style="list-style-type: none"> • MongoDB • Amazon DocumentDB
load-order	Integer positif. Nilai maksimumnya adalah 2,147,483,647.	Prioritas untuk memuat tabel dan tampilan. Tabel dan pandangan dengan nilai yang lebih tinggi dimuat pertama.

Parameter	Kemungkinan nilai	Deskripsi
<code>filters</code>	Susunan rangkaian.	Satu atau lebih objek untuk menyaring sumbernya. Anda menentukan parameter objek untuk memfilter pada satu kolom di sumber. Anda menentukan beberapa objek untuk memfilter pada beberapa kolom. Untuk informasi selengkapnya, lihat Menggunakan filter sumber .

Example Memigrasi semua tabel dalam skema

Contoh berikut bermigrasi semua tabel dari skema bernama `Test` di sumber Anda ke titik akhir target Anda.

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "Test",
        "table-name": "%"
      },
      "rule-action": "include"
    }
  ]
}
```

Example Memigrasi beberapa tabel dalam skema

Contoh berikut bermigrasi semua tabel kecuali yang dimulai dengan `DMS` dari sebuah skema bernama `Test` di sumber Anda ke titik akhir target Anda.

```
{
  "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 Memigrasi tabel tunggal tertentu dalam skema tunggal

Contoh berikut memigrasikan `Customer` tabel dari `NewCust` skema di sumber Anda ke titik akhir target Anda.

```

{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "NewCust",
        "table-name": "Customer"
      },
      "rule-action": "explicit"
    }
  ]
}

```

Note

Anda dapat secara eksplisit memilih pada beberapa tabel dan skema dengan menentukan beberapa aturan seleksi.

Example Memigrasi tabel dalam urutan set

Contoh berikut ini mengacu pada dua tabel. Tabel `loadfirst` (dengan prioritas 1) diinisialisasi sebelum `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 berlaku untuk inialisasi tabel. Beban tabel berturut-turut tidak akan menunggu beban tabel sebelumnya selesai jika `MaxFullLoadSubTasks` lebih besar dari 1.

Example Memigrasi beberapa tampilan dalam skema

Contoh berikut bermigrasi beberapa pandangan dari skema bernama `Test` di sumber Anda untuk tabel setara dalam target Anda.

```
{
  "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 Memigrasi semua tabel dan tampilan dalam skema

Contoh berikut bermigrasi semua tabel dan pandangan dari skema bernama `report` di sumber Anda untuk tabel setara dalam target Anda.

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "3",
      "rule-name": "3",
      "object-locator": {
        "schema-name": "report",
        "table-name": "%",

```

```

        "table-type": "all"
      },
      "rule-action": "include"
    }
  ]
}

```

Wildcard dalam pemetaan tabel

Bagian ini menjelaskan wildcard yang dapat Anda gunakan saat menentukan skema dan nama tabel untuk pemetaan tabel.

Wildcard	Pertandingan
%	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'.

Untuk sumber Oracle dan titik akhir target, Anda dapat menggunakan atribut koneksi `escapeCharacter` tambahan untuk menentukan karakter escape. Karakter escape memungkinkan Anda untuk menggunakan karakter wildcard tertentu dalam ekspresi seolah-olah itu tidak liar. Misalnya, `escapeCharacter=#` memungkinkan Anda untuk menggunakan '#' untuk membuat karakter wildcard bertindak sebagai karakter biasa dalam ekspresi seperti dalam kode contoh ini.

```

{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "542485267",

```

```
    "rule-name": "542485267",
    "object-locator": { "schema-name": "R00T", "table-name": "TEST#_T%" },
    "rule-action": "include",
    "filters": []
  }
]
```

Di sini, karakter escape '#' membuat karakter wildcard '_' bertindak sebagai karakter normal. AWS DMS memilih tabel dalam skema bernama R00T, di mana setiap tabel memiliki nama dengan TEST_T sebagai awalan.

Aturan dan tindakan transformasi

Anda menggunakan tindakan transformasi untuk menentukan setiap transformasi Anda ingin menerapkan skema yang dipilih, tabel, atau tampilan. Aturan transformasi adalah opsional.

Batasan

- Anda tidak dapat menerapkan lebih dari satu tindakan aturan transformasi terhadap objek yang sama (skema, tabel, kolom, table-tablespace, atau index-tablespace). Anda dapat menerapkan beberapa tindakan aturan transformasi pada tingkat apa pun selama setiap tindakan transformasi diterapkan terhadap objek yang berbeda.
- Nama tabel dan nama kolom dalam aturan transformasi peka huruf besar/kecil. Misalnya, Anda harus memberikan nama tabel dan nama kolom untuk database Oracle atau Db2 dalam huruf besar.
- Transformasi tidak didukung untuk nama kolom dengan bahasa kanan-ke-kiri.
- Transformasi tidak dapat dilakukan pada kolom yang berisi karakter khusus (misalnya #, \, /, -) dalam namanya.
- Satu-satunya transformasi yang didukung untuk kolom yang dipetakan ke tipe data BLOB/CLOB adalah dengan menjatuhkan kolom pada target.
- AWS DMS tidak mendukung replikasi dua tabel sumber ke satu tabel target. AWS DMS mereplikasi catatan dari tabel ke tabel, dan dari kolom ke kolom, sesuai dengan aturan transformasi tugas replikasi. Nama objek harus unik untuk mencegah tumpang tindih.

Misalnya, tabel sumber memiliki kolom bernama ID dan tabel target yang sesuai memiliki kolom yang sudah ada sebelumnya yang disebut id. Jika aturan menggunakan ADD-COLUMN pernyataan

untuk menambahkan kolom baru yang disebut `id`, dan pernyataan SQLite untuk mengisi kolom dengan nilai kustom, ini menciptakan duplikat, objek ambigu bernama `id` dan tidak didukung.

Nilai

Untuk aturan pemetaan tabel yang menggunakan jenis aturan transformasi, Anda dapat menerapkan nilai berikut.

Parameter	Kemungkinan nilai	Deskripsi
<code>rule-type</code>	<code>transformation</code>	Nilai yang berlaku aturan untuk setiap objek yang ditentukan oleh aturan seleksi. Gunakan <code>transformation</code> kecuali dinyatakan lain.
<code>rule-id</code>	Nilai numerik.	Nilai numerik yang unik untuk mengidentifikasi aturan. Jika Anda menentukan beberapa aturan transformasi untuk objek yang sama (skema, tabel, kolom, ruang antar-tabel, atau ruang tabel indeks), AWS DMS terapkan aturan transformasi dengan <code>rule-id</code> yang lebih rendah.
<code>rule-name</code>	Nilai alfanumerik.	Nama unik untuk mengidentifikasi aturan.
<code>object-locator</code>	Objek dengan parameter berikut: <ul style="list-style-type: none"> <code>schema-name</code> — Nama skema. Untuk titik akhir MongoDB dan Amazon DocumentDB, ini adalah nama database yang menyimpan satu set koleksi. <code>table-name</code> — Nama tabel, tampilan, atau koleksi. 	Nama setiap skema, tabel atau tampilan, tabel <code>tablespace</code> , <code>tablespace</code> indeks, dan kolom yang aturan berlaku. Anda dapat menggunakan tanda persen “%” sebagai wildcard untuk semua atau sebagian dari nilai di setiap parameter <code>object-locator</code> , kecuali <code>data-type</code> . Dengan demikian, Anda bisa mencocokkan barang-barang ini:

Parameter	Kemungkinan nilai	Deskripsi
	<ul style="list-style-type: none"> • <code>table-tablespace-name</code> — Nama tablespace tabel yang ada. • <code>index-tablespace-name</code> — Nama tablespace indeks yang ada. • <code>column-name</code> — Nama kolom yang ada. • <code>data-type</code> — Nama tipe data kolom yang ada. 	<ul style="list-style-type: none"> • Sebuah meja tunggal atau tampilan dalam skema tunggal • Sebuah tabel tunggal atau tampilan dalam beberapa atau semua skema • Beberapa atau semua tabel dan pandangan dalam skema tunggal • Beberapa atau semua tabel dan pandangan dalam beberapa atau semua skema • Satu atau lebih kolom dalam tabel tertentu atau tabel, pandangan atau pandangan, dan skema atau skema. • Kolom dengan <code>data-type</code> ketika beberapa kolom ditentukan. Untuk nilai yang mungkin <code>data-type</code>, lihat <code>data-type</code> dijelaskan berikut dalam tabel ini. <p>Juga, parameter <code>table-tablespace-name</code> atau <code>index-tablespace-name</code> ini hanya tersedia untuk mencocokkan titik akhir sumber Oracle. Anda dapat menentukan baik <code>table-tablespace-name</code> atau <code>index-tablespace-name</code> dalam satu aturan, tetapi tidak keduanya. Dengan demikian, Anda bisa mencocokkan baik item berikut:</p>

Parameter	Kemungkinan nilai	Deskripsi
		<ul style="list-style-type: none">• Satu, beberapa, atau semua tablespace meja• Satu, beberapa, atau semua tablespace indeks

Parameter	Kemungkinan nilai	Deskripsi
<code>rule-action</code>	<code>add-column</code> , <code>include-column</code> , <code>remove-column</code> <code>rename</code> <code>convert-lowercase</code> , <code>convert- uppercase</code> <code>add-prefix</code> , <code>remove-prefix</code> , <code>replace-prefix</code> <code>add-suffix</code> , <code>remove-suffix</code> , <code>replace-suffix</code> <code>define-primary-key</code> <code>change-data-type</code> <code>add-before-image-columns</code>	<p>Transformasi yang ingin Anda terapkan ke objek. Semua tindakan aturan transformasi peka huruf besar-kecil.</p> <p>Nilai <code>add-column</code> dari parameter <code>rule-action</code> menambahkan kolom ke tabel. Tetapi Anda tidak dapat menambahkan kolom baru dengan nama yang sama dengan kolom yang ada dari tabel yang sama.</p> <p>Ketika digunakan dengan <code>expression</code> dan <code>data-type</code> Parameter, <code>add-column</code> menentukan nilai data kolom baru.</p> <p>Nilai <code>change-data-type</code> untuk <code>rule-action</code> hanya tersedia untuk <code>column</code> target aturan.</p> <p>Nilai <code>include-column</code> dari parameter <code>rule-action</code> mengubah modus tabel ke jatuhkan semua kolom secara default dan sertakan kolom yang ditentukan. Beberapa kolom termasuk dalam target dengan menerapkan <code>include-column</code> aturan beberapa kali.</p> <p>Anda tidak dapat menggunakan <code>define-primary-key</code> aturan ketika aturan memiliki wildcard (%) dalam skema atau nama tabel.</p>

Parameter	Kemungkinan nilai	Deskripsi
		Untuk tugas yang ada, tindakan aturan transformasi yang mengubah skema tabel target seperti <code>remove-column</code> , <code>rename</code> , atau tidak <code>add-prefix</code> akan berlaku sampai Anda memulai ulang tugas. Jika Anda melanjutkan tugas setelah menambahkan aturan transformasi, Anda mungkin melihat perilaku tak terduga untuk kolom yang diubah, yang mungkin menyertakan data kolom yang hilang. Diperlukan restart tugas untuk memastikan aturan transformasi berfungsi dengan baik.
<code>rule-target</code>	<code>schema, table, column, table-tablespace, index-tablespace</code>	<p>Jenis objek yang Anda ubah.</p> <p>Parameter <code>table-tablespace</code> dan <code>index-tablespace</code> nilai hanya tersedia untuk Oracle target endpoint.</p> <p>Pastikan untuk menentukan nilai untuk parameter yang Anda tentukan sebagai bagian dari <code>object-locator</code>: <code>table-tablespace-name</code> atau <code>index-tablespace-name</code> nama.</p>
<code>value</code>	Nilai alfanumerik yang mengikuti aturan penamaan untuk jenis target.	Nilai baru untuk tindakan yang memerlukan masukan, seperti <code>rename</code> .
<code>old-value</code>	Nilai alfanumerik yang mengikuti aturan penamaan untuk jenis target.	Nilai lama untuk tindakan yang memerlukan penggantian, seperti <code>replace-prefix</code> .

Parameter	Kemungkinan nilai	Deskripsi
data-type	<p>type— Tipe data untuk digunakan jika rule-action adalah add-column atau tipe data pengganti jika rule-action adalah change-data-type .</p> <p>Atau, nama tipe data pengganti saat rule-action adalah change-data-type , nilai dari column-name adalah "%", dan parameter data-type tambahan untuk mengidentifikasi tipe data yang ada termasuk dalam object-locator .</p> <p>AWS DMS mendukung transformasi tipe data kolom untuk tipe data DMS berikut: "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"</p> <p>precision — Jika kolom ditambahkan atau tipe data pengganti memiliki presisi, nilai integer untuk menentukan presisi.</p> <p>scale— Jika kolom tambahan atau tipe data pengganti memiliki skala, nilai integer atau nilai waktu tanggal untuk menentukan skala.</p>	<p>Berikut ini adalah contoh parameter data-type untuk menentukan tipe data yang ada untuk diganti.</p> <pre data-bbox="974 394 1507 1747"> { "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": "" } }] } </pre> <p>Di sini, col10 kolom table_t tabel diubah menjadi tipe string data.</p>

Parameter	Kemungkinan nilai	Deskripsi
	length— Panjang data kolom baru (bila digunakan dengan <code>add-column</code>)	

Parameter	Kemungkinan nilai	Deskripsi
expression	Nilai alfanumerik yang mengikuti sintaks SQLite.	<p>Ketika digunakan dengan <code>rule-action</code> diatur ke <code>rename-schema</code>, <code>expression</code> parameter menentukan skema baru.</p> <p>Ketika digunakan dengan <code>rule-action</code> diatur ke <code>rename-table</code>, <code>expression</code> menentukan tabel baru. Ketika digunakan dengan <code>rule-action</code> diatur ke <code>rename-column</code>, <code>expression</code> menentukan nilai nama kolom baru.</p> <p>Ketika digunakan dengan <code>rule-action</code> diatur ke <code>add-column</code>, <code>expression</code> menentukan data yang membentuk kolom baru.</p> <p>Perhatikan bahwa hanya ekspresi yang didukung untuk parameter ini. Operator dan perintah tidak didukung.</p> <p>Untuk informasi selengkapnya tentang penggunaan ekspresi untuk aturan transformasi, lihat Menggunakan ekspresi aturan transformasi untuk menentukan isi kolom.</p> <p>Untuk informasi selengkapnya tentang ekspresi SQLite, lihat Menggunakan fungsi SQLite untuk membangun ekspresi.</p>

Parameter	Kemungkinan nilai	Deskripsi
<code>primary-key-def</code>	<p>Objek dengan parameter berikut:</p> <ul style="list-style-type: none"><code>name</code>— Nama kunci primer baru atau indeks unik untuk tabel atau tampilan.(Opsional)<code>origin</code>— Jenis kunci unik untuk mendefinisikan: <code>primary-key</code> (default) atau <code>unique-index</code> .<code>columns</code>— Sebuah array string daftar nama kolom dalam urutan mereka muncul dalam kunci primer atau indeks unik.	<p>Parameter ini dapat menentukan nama, jenis, dan isi dari kunci unik pada tabel atau tampilan yang diubah. Hal itu dilakukan ketika <code>rule-action</code> diatur ke <code>define-primary-key</code> dan <code>rule-target</code> diatur ke <code>table</code>. Secara default, kunci unik didefinisikan sebagai kunci primer.</p>

Parameter	Kemungkinan nilai	Deskripsi
before-image-def	<p>Objek dengan parameter berikut:</p> <ul style="list-style-type: none"> • <code>column-prefix</code> — Nilai ditambahkan ke nama kolom. Nilai default-nya adalah <code>BI_</code>. • <code>column-suffix</code> — Nilai ditambahkan ke nama kolom. Nilai default adalah (kosong). • <code>column-filter</code> Membutuhkan salah satu nilai berikut: <code>pk-only</code>(default), <code>non-lob</code>(opsional) dan <code>all</code>(opsional). 	<p>Parameter ini mendefinisikan konvensi penamaan untuk mengidentifikasi kolom citra sebelumnya dan menentukan filter untuk mengidentifikasi kolom sumber mana yang dapat memiliki kolom citra sebelumnya dibuat untuk mereka pada target. Anda dapat menentukan parameter ini ketika <code>rule-action</code> diatur ke <code>add-before-image-columns</code> dan <code>rule-target</code> diatur ke <code>column</code>.</p> <p>Jangan atur kedua <code>column-prefix</code> dan <code>column-suffix</code> untuk mengosongkan string.</p> <p>Untuk <code>column-filter</code>, pilih:</p> <ul style="list-style-type: none"> • <code>pk-only</code>— Untuk menambahkan hanya kolom yang merupakan bagian dari kunci primer tabel. • <code>non-lob</code>— Untuk menambahkan hanya kolom yang tidak tipe LOB. • <code>all</code>— Untuk menambahkan kolom yang memiliki nilai sebelum-gambar. <p>Untuk informasi lebih lanjut tentang dukungan sebelum-gambar untuk AWS DMS target endpoint, lihat:</p> <ul style="list-style-type: none"> • Menggunakan citra sebelumnya untuk melihat nilai asli dari baris

Parameter	Kemungkinan nilai	Deskripsi
		<p>CDC untuk Kinesis data stream sebagai target</p> <ul style="list-style-type: none"> • Menggunakan citra sebelumnya untuk melihat nilai-nilai asli dari baris CDC untuk Apache Kafka sebagai target

Contoh-contoh

Example Ubah Nama Skema

Contoh berikut mengganti nama skema dari `Test` di sumber Anda untuk `Test1` dalam target Anda.

```
{
  "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 Ubah Nama Tabel

Contoh berikut mengganti nama tabel dari `Actordi` sumber Anda untuk `Actor1` dalam target Anda.

```
{
  "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 Ubah Nama Kolom

Contoh berikut mengganti nama kolom dalam tabel `Actordi` dari `first_name` di sumber Anda untuk `fname` dalam target Anda.

```
{
  "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 Mengubah nama tablespace tabel Oracle

Contoh berikut mengganti nama tablespace tabel bernama `SetSpace` untuk sebuah tabel bernama `Actordi` di sumber Oracle Anda untuk `SceneTbl1Space` di titik akhir target Oracle Anda.

```

{
  "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 Mengubah nama sebuah tablespace indeks Oracle

Contoh berikut mengganti nama tablespace indeks bernama SetISpace untuk sebuah tabel bernama Actor di sumber Oracle Anda untuk SceneIdxSpace di titik akhir target Oracle Anda.

```

{
  "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 Tambahkan kolom

Contoh berikut menambahkan datetime kolom ke tabel Actor dalam skema 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 Hapus kolom

Contoh berikut mengubah tabel bernama Actor di sumber Anda untuk menghapus semua kolom yang dimulai dengan karakter c dari itu di target Anda.

```
{
  "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 Mengkonversi ke lowercase

Contoh berikut mengkonversi nama tabel dari `ACTOR` di sumber Anda untuk `actord` dalam target Anda.

```

{
  "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 Mengkonversi ke huruf besar

Contoh berikut mengkonversi semua kolom di semua tabel dan semua skema dari huruf kecil di sumber Anda untuk huruf besar dalam target Anda.

```
{  
  "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 Tambahkan prefiks

Contoh berikut mengubah semua tabel di sumber Anda untuk menambahkan awalanDMS_ kepada mereka dalam target Anda.

```
{
```

```

"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 Ganti awalan

Contoh berikut mengubah semua kolom yang berisi awalanPre_di sumber anda untuk mengganti awalan denganNewPre_dalam target Anda.

```

{
  "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 Hapus sufiks

Contoh berikut mengubah semua tabel di sumber Anda untuk menghapus akhiran `_DMS` dari mereka di target Anda.

```

{
  "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 Mendefinisikan kunci primer

Contoh berikut mendefinisikan kunci primer bernama `ITEM-primary-key` pada tiga kolom `ITEM` tabel bermigrasi ke titik akhir target Anda.

```
{
  "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 Tentukan indeks yang unik

Contoh berikut mendefinisikan indeks unik bernama `ITEM-unique-idx` pada tiga kolom `ITEM` tabel bermigrasi ke titik akhir target Anda.

```
{
  "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 Mengubah tipe data dari kolom target

Contoh berikut mengubah tipe data dari kolom target bernama `SALE_AMOUNT` dari tipe data yang ada ke `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 Tambahkan kolom gambar sebelum

Untuk kolom sumber bernamaemp_no, aturan transformasi dalam contoh berikut menambahkan kolom baru bernamaBI_emp_notarget.

```

{
  "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"
    }
  }
  ]
}

```

Di sini, pernyataan berikut populates `BI_emp_nokolom` di baris yang sesuai dengan 1.

```
UPDATE employees SET emp_no = 3 WHERE BI_emp_no = 1;
```

Saat menulis pembaruan CDC ke AWS DMS target yang didukung, `BI_emp_no` kolom memungkinkan untuk mengetahui baris mana yang memiliki nilai terbaru di `emp_no` kolom.

Menggunakan ekspresi aturan transformasi untuk menentukan isi kolom

Untuk menentukan konten untuk kolom baru dan yang sudah ada, Anda dapat menggunakan ekspresi dalam aturan transformasi. Misalnya, menggunakan ekspresi Anda dapat menambahkan kolom atau mereplikasi header tabel sumber ke target. Anda juga dapat menggunakan ekspresi untuk menandai catatan pada tabel target sebagai dimasukkan, diperbarui, atau dihapus pada sumber.

Topik

- [Menambahkan kolom menggunakan ekspresi](#)
- [Menandai catatan target menggunakan ekspresi](#)
- [Replikasi header tabel sumber menggunakan ekspresi](#)
- [Menggunakan fungsi SQLite untuk membangun ekspresi](#)
- [Menambahkan metadata ke tabel target menggunakan ekspresi](#)

Menambahkan kolom menggunakan ekspresi

Untuk menambahkan kolom ke tabel menggunakan ekspresi dalam aturan transformasi, gunakan `add-column` aturan tindakan dan `columnTarget` aturan.

Contoh berikut menambahkan kolom baru ke `ITEM` tabel. Ini menetapkan nama kolom baru untuk `FULL_NAME`, dengan tipe `datastring`, 50 karakter panjang. Ekspresi menggabungkan nilai-nilai dari dua kolom yang ada, `FIRST_NAME` dan `LAST_NAME`, untuk mengevaluasi `FULL_NAME`. Parameter `schema-name`, `table-name`, dan parameter ekspresi mengacu pada objek dalam tabel database sumber. `Value` dan `data-type` mengacu pada obyek dalam tabel basis data target.

```
{
  "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
    }
  }
]
}

```

Menandai catatan target menggunakan ekspresi

Untuk menandai catatan dalam tabel target sebagai dimasukkan, diperbarui, atau dihapus dalam tabel sumber, menggunakan ekspresi dalam aturan transformasi. Ekspresi menggunakan `operation_indicator` fungsi untuk menandai catatan. Rekaman yang dihapus dari sumber tidak dihapus dari target. Sebaliknya, data target ditandai dengan nilai yang disediakan pengguna untuk menunjukkan bahwa itu telah dihapus dari sumber.

Note

`operation_indicator` Fungsi ini hanya berfungsi pada tabel yang memiliki kunci utama pada basis data sumber dan target.

Sebagai contoh, aturan transformasi berikut pertama menambahkan baru `Operation` kolom ke tabel target. Kemudian update kolom dengan nilai `D` setiap kali catatan dihapus dari tabel sumber.

```
{
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "rule-target": "column",
  "object-locator": {
    "schema-name": "%",
    "table-name": "%"
  },
  "rule-action": "add-column",
  "value": "0peration",
  "expression": "operation_indicator('D', 'U', 'I')",
  "data-type": {
    "type": "string",
    "length": 50
  }
}
```

Replikasi header tabel sumber menggunakan ekspresi

Secara default, header untuk tabel sumber tidak direplikasi ke target. Untuk menunjukkan header untuk mereplikasi, menggunakan aturan transformasi dengan ekspresi yang mencakup header kolom tabel.

Anda dapat menggunakan header kolom berikut dalam ekspresi.

Header	Nilai dalam replikasi yang sedang berlangsung	Nilai dalam beban penuh	Tipe Data
AR_H_STRE AM_POSITION	Nilai posisi aliran dari sumber. Nilai ini mungkin nomor perubahan sistem (SCN) atau nomor urutan log (LSN),	String kosong.	STRING

Header	Nilai dalam replikasi yang sedang berlangsung	Nilai dalam beban penuh	Tipe Data
	tergantung pada titik akhir sumber.		
AR_H_TIMESTAMP	Sebuah timestamp yang menunjukkan waktu perubahan.	Sebuah timestamp menunjukkan data waktu saat ini tiba di target.	DATETIME (skala = 7)
AR_H_COMMIT_TIMESTAMP	Sebuah timestamp menunjukkan waktu komit.	Sebuah timestamp menunjukkan waktu saat ini.	DATETIME (skala = 7)
AR_H_OPERATION	INSERT, UPDATE	SISIPAN	STRING
PENGGUNA	Nama pengguna, ID, atau informasi lain yang sumber menyediakan tentang pengguna yang membuat perubahan. Header ini didukung pada SQL Server dan Oracle (versi 11.2.0.3 dan lebih tinggi) sumber endpoint hanya.	Transformasi yang ingin Anda terapkan ke objek. Tindakan aturan transformasi peka huruf besar kecil.	STRING

Header	Nilai dalam replikasi yang sedang berlangsung	Nilai dalam beban penuh	Tipe Data
AR_H_CHANNEL_SEQ	Nomor incrementing unik dari database sumber yang terdiri dari timestamp dan nomor auto incrementing. Nilai tergantung pada sistem database sumber.	String kosong.	STRING

Contoh berikut menambahkan kolom baru ke target dengan menggunakan nilai posisi aliran dari sumber. Untuk SQL Server, nilai posisi aliran adalah LSN untuk titik akhir sumber. Untuk Oracle, nilai posisi aliran adalah SCN untuk titik akhir sumber.

```
{
  "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
  }
}
```

Contoh berikut menambahkan kolom baru ke target yang memiliki nomor incrementing unik dari sumber. Nilai ini mewakili 35 digit nomor unik di tingkat tugas. 16 digit pertama adalah bagian dari timestamp, dan 19 digit terakhir adalah jumlah record_id bertambah oleh DBMS.

```
{
  "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
  }
}
```

Menggunakan fungsi SQLite untuk membangun ekspresi

Anda menggunakan pengaturan tabel untuk menentukan pengaturan yang ingin Anda terapkan ke tabel yang dipilih atau tampilan untuk operasi tertentu. Aturan pengaturan meja bersifat opsional.

Note

Alih-alih konsep tabel dan pandangan, MongoDB dan database DocumentDB menyimpan catatan data sebagai dokumen yang berkumpul bersama-sama diKoleksi. Jadi kemudian, ketika bermigrasi dari MongoDB atau sumber DocumentDB, mempertimbangkan jenis rentang segmentasi pengaturan beban paralel untuk dipilihKoleksibukan tabel dan pandangan.

Topik

- [Menggunakan ekspresi CASE](#)
- [Contoh-contoh](#)

Berikut ini, Anda dapat menemukan fungsi string yang dapat Anda gunakan untuk membangun ekspresi aturan transformasi.

Fungsi string	Deskripsi
<code>lower(x)</code>	<code>lower(x)</code> Fungsi mengembalikan salinan string <code>x</code> dengan semua karakter dikonversi ke huruf kecil. <code>lower</code> Fungsi bawaan hanya berfungsi untuk karakter ASCII.
<code>upper(x)</code>	<code>upper(x)</code> Fungsi mengembalikan salinan string <code>x</code> dengan semua karakter dikonversi ke huruf besar. <code>upper</code> Fungsi bawaan hanya berfungsi untuk karakter ASCII.
<code>ltrim(x,y)</code>	<code>ltrim(x,y)</code> Fungsi mengembalikan string yang dibentuk dengan menghapus semua karakter yang muncul di <code>y</code> dari sisi kiri <code>x</code> . Jika tidak ada nilai untuk <code>y</code> , <code>ltrim(x)</code> hapus spasi dari sisi kiri <code>x</code> .
<code>replace(x,y,z)</code>	<code>replace(x,y,z)</code> Fungsi mengembalikan string dibentuk dengan mengganti string <code>z</code> untuk setiap kejadian string <code>y</code> dalam string <code>x</code> .
<code>rtrim(x,y)</code>	<code>rtrim(x,y)</code> Fungsi mengembalikan string yang dibentuk dengan menghapus semua karakter yang muncul di <code>y</code> dari sisi kanan <code>x</code> . Jika tidak ada nilai untuk <code>y</code> , <code>rtrim(x)</code> hapus spasi dari sisi kanan <code>x</code> .
<code>substr(x,y,z)</code>	<p><code>substr(x,y,z)</code> Fungsi mengembalikan substring dari string input <code>x</code> yang dimulai dengan karakter <code>y</code> th, dan yang merupakan <code>z</code> karakter panjang.</p> <p>Jika <code>z</code> dihilangkan, <code>substr(x,y)</code> mengembalikan semua karakter melalui akhir string <code>x</code> dimulai dengan karakter <code>y</code> th. Karakter paling kiri <code>x</code> adalah nomor 1. Jika <code>y</code> negatif, karakter pertama substring ditemukan dengan menghitung dari kanan daripada kiri. Jika <code>z</code> negatif, maka <code>abs(z)</code> karakter yang mendahului karakter ke <code>y</code> - dikembalikan. Jika <code>x</code> adalah string, maka indeks karakter mengacu pada karakter UTF-8 yang sebenarnya. Jika <code>x</code> adalah BLOB, maka indeks mengacu pada byte.</p>

Fungsi string	Deskripsi
<code>trim(x,y)</code>	<code>trim(x,y)</code> Fungsi mengembalikan string dibentuk dengan menghapus semua karakter yang muncul di <code>y</code> dari kedua sisi <code>x</code> . Jika tidak ada nilai untuk <code>y</code> , <code>trim(x)</code> menghapus spasi dari kedua sisi <code>x</code> .

Berikut ini, Anda dapat menemukan fungsi LOB yang dapat Anda gunakan untuk membangun ekspresi aturan transformasi.

Fungsi LOB	Deskripsi
<code>hex(x)</code>	<code>hex</code> Fungsi menerima BLOB sebagai argumen dan mengembalikan versi string heksadesimal huruf besar dari konten BLOB.
<code>randblob (N)</code>	<code>randblob(N)</code> Fungsi mengembalikan BLOB <code>N</code> -byte yang berisi byte pseudorandom. Jika <code>N</code> kurang dari 1, BLOB acak 1-byte dikembalikan.
<code>zeroblob(N)</code>	<code>zeroblob(N)</code> Fungsi mengembalikan BLOB yang terdiri dari <code>N</code> byte 0x00.

Berikut ini, Anda dapat menemukan fungsi numerik yang dapat Anda gunakan untuk membangun ekspresi aturan transformasi.

Fungsi numerik	Deskripsi
<code>abs(x)</code>	<code>abs(x)</code> Fungsi mengembalikan nilai absolut dari argumen <code>x</code> numerik. <code>abs(x)</code> Fungsi mengembalikan NULL jika <code>x</code> adalah NULL. <code>abs(x)</code> Fungsi mengembalikan 0.0 jika <code>x</code> adalah string atau BLOB yang tidak dapat dikonversi ke nilai numerik.
<code>random()</code>	<code>random</code> Fungsi mengembalikan bilangan bulat pseudorandom antara -9,223,372,036,854,775.808 dan +9,223,372,036,854.775,807.

Fungsi numerik	Deskripsi
<code>round (x,y)</code>	<code>round (x,y)</code> Fungsi mengembalikan nilai floating-point <code>x</code> dibulatkan ke digit <code>y</code> di sebelah kanan titik desimal. Jika tidak ada nilai untuk <code>y</code> , itu diasumsikan 0.
<code>max (x,y...)</code>	<code>max</code> Fungsi multiargument mengembalikan argumen dengan nilai maksimum, atau mengembalikan NULL jika argumen adalah NULL. <code>max</code> Fungsi mencari argumennya dari kiri ke kanan untuk argumen yang mendefinisikan fungsi penyusunan. Jika ditemukan, ia menggunakan fungsi penyusunan itu untuk semua perbandingan string. Jika tidak ada argumen untuk <code>max</code> mendefinisikan fungsi penyusunan, fungsi BINARY penyusunan digunakan. <code>max</code> Fungsi adalah fungsi sederhana ketika memiliki dua atau lebih argumen, tetapi beroperasi sebagai fungsi agregat jika memiliki argumen tunggal.
<code>min (x,y...)</code>	<code>min</code> Fungsi multiargument mengembalikan argumen dengan nilai minimum. <code>min</code> Fungsi mencari argumennya dari kiri ke kanan untuk argumen yang mendefinisikan fungsi penyusunan. Jika ditemukan, ia menggunakan fungsi penyusunan itu untuk semua perbandingan string. Jika tidak ada argumen untuk <code>min</code> mendefinisikan fungsi penyusunan, fungsi BINARY penyusunan digunakan. <code>min</code> Fungsi adalah fungsi sederhana ketika memiliki dua atau lebih argumen, tetapi beroperasi sebagai fungsi agregat jika memiliki argumen tunggal.

Berikut ini, Anda dapat menemukan fungsi pemeriksaan NULL yang dapat Anda gunakan untuk membangun ekspresi aturan transformasi.

Fungsi pemeriksaan NULL	Deskripsi
<code>coalesce (x,y...)</code>	<code>coalesce</code> Fungsi mengembalikan salinan argumen non-Null pertama, tetapi mengembalikan NULL jika semua argumen adalah NULL. Fungsi <code>coalesce</code> memiliki setidaknya dua argumen.
<code>ifnull(x,y)</code>	<code>ifnull</code> Fungsi mengembalikan salinan argumen non-Null pertama, tetapi mengembalikan NULL jika kedua argumen adalah NULL. <code>ifnull</code> Fungsi ini memiliki dua argumen. <code>ifnull</code> Fungsinya sama <code>coalesce</code> dengan dua argumen.
<code>nullif(x,y)</code>	<p><code>nullif(x,y)</code> Fungsi mengembalikan salinan argumen pertama jika argumen yang berbeda, tetapi mengembalikan NULL jika argumen yang sama.</p> <p><code>nullif(x,y)</code> Fungsi mencari argumennya dari kiri ke kanan untuk argumen yang mendefinisikan fungsi penyusunan. Jika ditemukan, ia menggunakan fungsi penyusunan itu untuk semua perbandingan string. Jika tidak ada argumen untuk <code>nullif</code> mendefinisikan fungsi penyusunan, maka fungsi penyusunan digunakan. BINARY</p>

Berikut ini, Anda dapat menemukan fungsi tanggal dan waktu yang dapat Anda gunakan untuk membangun ekspresi aturan transformasi.

Fungsi tanggal dan waktu	Deskripsi
<code>date(timestring , modifier, modifier...)</code>	<code>date</code> Fungsi mengembalikan tanggal dalam format YYYY-MM-DD.
<code>time(timestring , modifier, modifier...)</code>	<code>time</code> Fungsi mengembalikan waktu dalam format HH: MM: SS.
<code>datetime(timestring , modifier, modifier...)</code>	<code>datetime</code> Fungsi mengembalikan tanggal dan waktu dalam format YYYY-MM-DD HH: MM: SS.

Fungsi tanggal dan waktu	Deskripsi
<code>julianday(<i>timestring</i> , <i>modifier</i> , <i>modifier</i>...)</code>	<p><code>julianday</code> Fungsi mengembalikan jumlah hari sejak siang di Greenwich pada 24 November 4714 SM.</p>
<code>strftime(<i>format</i> , <i>timestring</i> , <i>modifier</i> , <i>modifier</i>...)</code>	<p><code>strftime</code> Fungsi mengembalikan tanggal sesuai dengan format string ditentukan sebagai argumen pertama, menggunakan salah satu variabel berikut:</p> <ul style="list-style-type: none"> <code>%d</code>: hari bulan <code>%H</code>: jam 00—24 <code>%f</code>: ** pecahan detik SS.SSS <code>%j</code>: hari tahun 001—366 <code>%J</code>: ** Nomor hari Julian <code>%m</code>: bulan 01—12 <code>%M</code>: menit 00—59 <code>%s</code>: detik sejak 1970-01-01 <code>%S</code>: detik 00—59 <code>%w</code>: hari dalam seminggu 0—6 Minggu==0 <code>%W</code>: minggu tahun 00—53 <code>%Y</code>: tahun 0000—9999 <code>%%</code>: %

Berikut ini, Anda dapat menemukan fungsi hash yang dapat Anda gunakan untuk membangun ekspresi aturan transformasi.

Fungsi hash	Deskripsi
hash_sha256(<i>x</i>)	<p>hashFungsi ini menghasilkan nilai hash untuk kolom input (menggunakan algoritma SHA-256) dan mengembalikan nilai heksadesimal dari nilai hash yang dihasilkan.</p> <p>Untuk menggunakan hash fungsi dalam ekspresi, tambahkan hash_sha256(<i>x</i>) ke ekspresi dan ganti <i>x</i> dengan nama kolom sumber.</p>

Menggunakan ekspresi CASE

CASE Ekspresi SQLite mengevaluasi daftar kondisi dan mengembalikan ekspresi berdasarkan hasil. Sintaks ditampilkan sebagai berikut.

```

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

```

Contoh-contoh

Example menambahkan kolom string baru ke tabel target menggunakan kondisi kasus

Contoh aturan transformasi berikut menambahkan kolom string baru, emp_seniority, ke tabel target, employee. Ini menggunakan round fungsi SQLite pada kolom gaji, dengan kondisi kasus untuk memeriksa apakah gaji sama atau melebihi 20.000. Jika ya, kolom mendapatkan nilai SENIOR, dan hal lain memiliki nilai 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 menambahkan kolom tanggal baru ke tabel target

Contoh berikut menambahkan kolom tanggal baru, `createdate`, ke tabel target, `employee`. Saat Anda menggunakan fungsi tanggal SQL `datedatetime`, tanggal ditambahkan ke tabel yang baru dibuat untuk setiap baris yang disisipkan.

```
{
  "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 menambahkan kolom numerik baru ke tabel target

Contoh berikut menambahkan kolom numerik baru, `rounded_emp_salary`, ke tabel target, `employee`. Ini menggunakan `round` fungsi SQLite untuk menambahkan gaji bulat.

```
{
  "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 menambahkan kolom string baru ke tabel target menggunakan fungsi hash

Contoh berikut menambahkan kolom string baru, `hashed_emp_number`, ke tabel target, `employee`. `hash_sha256(x)` Fungsi SQLite menciptakan nilai hash pada target untuk kolom sumber, `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
    }
  }
}

```

Menambahkan metadata ke tabel target menggunakan ekspresi

Anda dapat menambahkan informasi metadata ke tabel target dengan menggunakan ekspresi berikut:

- `$AR_M_SOURCE_SCHEMA`— Nama skema sumber.
- `$AR_M_SOURCE_TABLE_NAME`— Nama tabel sumber.
- `$AR_M_SOURCE_COLUMN_NAME`— Nama kolom di tabel sumber.
- `$AR_M_SOURCE_COLUMN_DATATYPE`— Tipe data kolom dalam tabel sumber.

Example menambahkan kolom untuk nama skema menggunakan nama skema dari sumber

Contoh berikut menambahkan kolom baru bernama `schema_name` target dengan menggunakan nama skema dari sumber.

```

{
  "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
  }
}

```

Tabel dan koleksi pengaturan aturan dan operasi

Gunakan pengaturan tabel untuk menentukan pengaturan apa pun yang ingin Anda terapkan ke tabel atau tampilan yang dipilih untuk operasi tertentu. Aturan pengaturan tabel bersifat opsional, tergantung pada titik akhir dan persyaratan migrasi Anda.

Alih-alih menggunakan tabel dan tampilan, database MongoDB dan Amazon DocumentDB menyimpan catatan data sebagai dokumen yang dikumpulkan bersama dalam koleksi. Database tunggal untuk setiap titik akhir MongoDB atau Amazon DocumentDB adalah kumpulan koleksi tertentu yang diidentifikasi dengan nama database.

Saat bermigrasi dari sumber MongoDB atau Amazon DocumentDB, Anda bekerja dengan pengaturan beban paralel sedikit berbeda. Dalam hal ini, pertimbangkan jenis autosegmentasi atau segmentasi rentang pengaturan beban paralel untuk koleksi yang dipilih daripada tabel dan tampilan.

Topik

- [Wildcard dalam pengaturan tabel dibatasi](#)
- [Menggunakan beban paralel untuk tabel, tampilan, dan koleksi yang dipilih](#)
- [Menentukan pengaturan LOB untuk tabel yang dipilih atau tampilan](#)
- [Contoh pengaturan tabel](#)

Untuk aturan pemetaan tabel yang menggunakan jenis aturan tabel-pengaturan, Anda dapat menerapkan parameter berikut.

Parameter	Kemungkinan nilai	Deskripsi
<code>rule-type</code>	<code>table-settings</code>	Nilai yang menerapkan aturan ke tabel, tampilan, atau koleksi yang ditentukan oleh aturan pemilihan.
<code>rule-id</code>	Nilai numerik.	Nilai numerik yang unik untuk mengidentifikasi aturan.
<code>rule-name</code>	Nilai alfanumerik.	Nama unik untuk mengidentifikasi aturan.

Parameter	Kemungkinan nilai	Deskripsi
object-locator	<p>Objek dengan parameter berikut:</p> <ul style="list-style-type: none">• <code>schema-name</code> — Nama skema. Untuk titik akhir MongoDB dan Amazon DocumentDB, ini adalah nama database yang menyimpan satu set koleksi.• <code>table-name</code> — Nama tabel, tampilan, atau koleksi.	Nama skema dan tabel atau tampilan tertentu atau nama database dan koleksi tertentu (tidak ada wildcard).

Parameter	Kemungkinan nilai	Deskripsi
parallel-load	<p>Objek dengan parameter berikut:</p> <ul style="list-style-type: none"> • type- Menentukan apakah pemuatan paralel dihidupkan. <p>Jika ya, parameter ini juga menentukan mekanisme untuk mengidentifikasi tabel atau tampilan partisi, subpartisi, atau segmen lain untuk memuat secara paralel. Partisi adalah segmen yang sudah didefinisikan dan diidentifikasi dengan nama dalam tabel sumber atau tampilan.</p> <p>Untuk titik akhir MongoDB dan Amazon DocumentDB, partisi adalah segmen. AWS DMS dapat menghitung ini secara otomatis diberikan parameter autosegmentasi terkait. Atau Anda dapat menentukan ini secara manual menggunakan parameter segmentasi rentang.</p> <p>Untuk Oracle endpoint saja, subpartisi adalah tingkat tambahan segmen yang sudah didefinisikan dan diidentifikasi oleh nama dalam tabel sumber atau tampilan. Anda dapat mengidentifikasi segmen lain di table-settings aturan dengan menentukan batas-batas dalam kisaran nilai untuk satu atau lebih tabel atau tampilan kolom.</p>	<p>Nilai yang menentukan beban paralel (multithreaded) operasi di atas meja atau tampilan diidentifikasi oleh object-locator Pilihan. Dalam kasus ini, Anda dapat memuat secara paralel dengan cara berikut:</p> <ul style="list-style-type: none"> • Dengan segmen yang ditentukan oleh semua partisi atau subpartisi yang tersedia. • Dengan partisi dan subpartisi yang dipilih. • Dengan autosegmentasi atau segmen berbasis rentang yang Anda tentukan. <p>Untuk informasi lebih lanjut tentang beban paralel, lihat Menggunakan beban paralel untuk tabel, tampilan, dan koleksi yang dipilih.</p>

Parameter	Kemungkinan nilai	Deskripsi
	<ul style="list-style-type: none"> • <code>partition</code> s Saattypeadalahpartition s-list , nilai ini menentuka n semua partisi untuk memuat secara paralel. • <code>subpartitions</code> — Untuk Endpoint Oracle saja, ketikatypeadalahpartitions- listnilai ini menentukan semua subpartisi untuk memuat secara paralel. • <code>columns</code>Saattypeadalahranges, nilai ini menentukan nama-nama kolom yang digunakan untuk mengidentifikasi segmen berbasis rentang untuk memuat secara paralel. • <code>boundarie</code> s Saattypeadalahranges, nilai ini menentukan nilai-nil aicolumnsdigunakan untuk mengidentifikasi segmen berbasis rentang untuk dimuat secara paralel. 	

Parameter	Kemungkinan nilai	Deskripsi
type	<p>Salah satu dari berikut ini untuk <code>parallel-load</code> :</p> <ul style="list-style-type: none"> • <code>partitions-auto</code> — Semua partisi dari tabel atau tampilan dimuat secara paralel. Setiap partisi dialokasikan untuk thread sendiri. <p>Ini adalah pengaturan yang diperlukan untuk titik akhir sumber MongoDB dan Amazon DocumentDB untuk menggunakan opsi autosegmentasi dari beban penuh paralel.</p> <ul style="list-style-type: none"> • <code>subpartitions-auto</code> — (Oracle endpoint saja) Semua subpartisi dari tabel atau tampilan dimuat secara paralel. Setiap subpartisi dialokasikan untuk thread sendiri. • <code>partitions-list</code> — Semua partisi yang ditentukan dari tabel atau tampilan dimuat secara paralel. Untuk Endpoint Oracle saja, semua subpartisi yang ditentukan dari tabel atau tampilan dimuat secara paralel. Setiap partisi dan subpartisi yang Anda tentukan dialokasikan untuk thread sendiri. Anda menentukan partisi dan subpartisi untuk dimuat secara paralel dengan nama partisi (<code>partitions</code>) dan nama subpartisi (<code>subpartitions</code>). 	<p>Mekanisme untuk mengidentifikasi tabel, tampilan, atau koleksi partisi, subpartisi, atau segmen untuk memuat secara paralel.</p>

Parameter	Kemungkinan nilai	Deskripsi
	<ul style="list-style-type: none"> • <code>ranges</code>— Semua segmen tabel, tampilan, atau koleksi yang ditentukan rentang dimuat secara paralel. Setiap tabel, tampilan, atau segmen koleksi yang Anda identifikasi dialokasikan ke utasnya sendiri. Anda menentukan segmen ini dengan nama kolom (<code>columns</code>) dan nilai kolom (<code>boundaries</code>). <p>PostgreSQL endpoint hanya mendukung jenis beban paralel ini. MongoDB dan Amazon DocumentDB sebagai titik akhir sumber mendukung jenis segmentasi rentang ini dan tipe autosegmentasi dari beban penuh paralel (<code>partitions-auto</code>).</p> <ul style="list-style-type: none"> • <code>none</code>— Tabel, tampilan, atau koleksi dimuat dalam tugas single-threaded (default), terlepas dari partisi atau subpartisi. Untuk informasi selengkapnya, lihat Membuat tugas. 	
<code>number-of-partitions</code>	(Opsional) type <code>partitions-auto</code> untuk koleksi tertentu dari titik akhir MongoDB atau Amazon DocumentDB, parameter ini menentukan jumlah total partisi (segmen) yang digunakan untuk migrasi. Defaultnya adalah 16.	Menentukan jumlah yang tepat dari partisi untuk memuat secara paralel.

Parameter	Kemungkinan nilai	Deskripsi
<code>collection-count-from-metadata</code>	(Opsional) type Kapan <code>partition s-auto</code> untuk koleksi tertentu dari titik akhir MongoDB atau Amazon DocumentDB dan parameter ini diatur <code>true</code> ke AWS DMS , menggunakan perkiraan jumlah koleksi untuk menentukan jumlah partisi. Jika parameter ini disetel ke <code>false</code> , AWS DMS gunakan jumlah koleksi yang sebenarnya. Nilai default-nya <code>true</code> .	Menentukan apakah akan menggunakan perkiraan jumlah koleksi atau jumlah koleksi aktual untuk menghitung jumlah partisi untuk memuat secara paralel.
<code>max-records-skip-per-page</code>	(Opsional) type Kapan <code>partition s-auto</code> untuk koleksi tertentu dari titik akhir MongoDB atau Amazon DocumentDB, ini adalah jumlah catatan yang harus dilewati sekaligus saat menentukan batas untuk setiap partisi. AWS DMS menggunakan pendekatan <code>skip paginated</code> untuk menentukan batas minimum untuk partisi. Default adalah 10.000.	Menentukan jumlah catatan untuk melewati sekaligus ketika menentukan batas-batas untuk setiap partisi. Menyetel nilai yang relatif besar dari default dapat mengakibatkan batas waktu kursor dan kegagalan tugas. Menetapkan nilai yang relatif rendah dari default menghasilkan lebih banyak operasi per halaman dan beban penuh yang lebih lambat.

Parameter	Kemungkinan nilai	Deskripsi
<code>batch-size</code>	(Opsional) <code>type</code> Kapan <code>partition s-auto</code> untuk koleksi tertentu dari titik akhir MongoDB atau Amazon DocumentDB, nilai integer ini membatasi jumlah dokumen yang dikembalikan dalam satu batch pulang-pergi. Jika ukuran batch nol (0), kursor menggunakan ukuran batch maksimum yang ditentukan server. Default-nya adalah 0.	Menentukan jumlah maksimum dokumen yang dikembalikan dalam satu batch. Setiap batch membutuhkan perjalanan pulang pergi ke server.
<code>partitions</code>	Saat <code>type</code> adalah <code>partitions-list</code> , ini adalah array string yang menentukan nama partisi untuk dimuat secara paralel.	Nama-nama partisi untuk dimuat secara paralel.
<code>subpartitions</code>	(Endpoint Oracle saja) Kapan <code>type</code> adalah <code>partitions-list</code> , ini adalah array string yang menentukan nama subpartisi untuk dimuat secara paralel.	Nama-nama subpartisi untuk dimuat secara paralel.
<code>columns</code>	<code>type</code> Kapan <code>ranges</code> , array string diatur ke nama kolom yang mengidentifikasi tabel berbasis rentang, tampilan, atau segmen koleksi untuk dimuat secara paralel.	Nama-nama kolom yang mengidentifikasi tabel berbasis rentang, tampilan, atau segmen koleksi untuk dimuat secara paralel.

Parameter	Kemungkinan nilai	Deskripsi
<p><code>boundaries</code></p>	<p>Saattypeadalahranges, array array kolom-nilai. Setiap array kolom-nilai berisi nilai-nilai kolom dalam kuantitas dan urutan yang ditentukan olehcolumns. Sebuah larik kolom-nilai menentukan batas atas tabel, tampilan, atau segmen koleksi. Setiap larik nilai kolom tambahan menambahkan batas atas untuk satu tabel tambahan, tampilan, atau segmen koleksi. Semua tabel, tampilan, atau segmen koleksi berbasis rentang tersebut dimuat secara paralel.</p>	<p>Nilai kolom yang mengidentifikasi tabel berbasis rentang, tampilan, atau partisi koleksi untuk dimuat secara paralel.</p>
<p><code>lob-settings</code></p>	<p>Objek dengan parameter berikut:</p> <ul style="list-style-type: none"> • <code>mode</code>— Menentukan modus penanganan migrasi untuk LOB. • <code>bulk-max-size</code> — Menentukan ukuran maksimum LOB, tergantung padamodePengaturan. 	<p>Nilai yang menentukan LOB penanganan untuk tabel atau tampilan diidentifikasi olehobject-locator Pilihan. Penanganan LOB ditentukan menimpa setiap pengaturan LOB tugas untuk tabel ini atau hanya melihat. Untuk informasi selengkapnya tentang penggunaan parameter pengaturan LOB, lihatMenentukan pengaturan LOB untuk tabel yang dipilih atau tampilan.</p>

Parameter	Kemungkinan nilai	Deskripsi
mode	<p>Menentukan penanganan migrasi untuk LOB dalam tabel tertentu atau tampilan menggunakan nilai berikut:</p> <ul style="list-style-type: none"> • <code>limited</code>— (Default) Nilai ini menetapkan migrasi ke mode LOB terbatas, dengan semua LOB bermigrasi inline bersama-sama dengan semua jenis data kolom lainnya dalam tabel atau tampilan. Gunakan nilai ini ketika mereplikasi LOB yang sebagian besar kecil (100 MB atau kurang). Juga, tentukan <code>bulk-max-size</code> nilai (nol tidak valid). Semua LOB bermigrasi lebih besar dari <code>bulk-max-size</code> dipotong ke ukuran yang Anda tetapkan. • <code>unlimited</code> — Nilai ini menetapkan migrasi ke modus LOB penuh. Gunakan nilai ini ketika semua atau sebagian besar LOB yang ingin Anda mereplikasi lebih besar dari 1 GB. Jika Anda menentukan nilai nol di <code>bulk-max-size</code> , semua LOB bermigrasi di Mode LOB penuh Standar. Dalam bentuk <code>unlimited mode</code>, semua LOB dimigrasi secara terpisah dari jenis data kolom lain menggunakan pencarian dari tabel sumber atau tampilan. Jika Anda menentukan <code>bulk-max-</code> 	Mekanisme yang digunakan untuk migrasi LOB.

Parameter	Kemungkinan nilai	Deskripsi
	<p>size nilai lebih besar dari nol, semua LOB dimigrasi diKombinasi Mode LOB penuh. Dalam bentuk mode unlimited , LOB yang lebih besar dari bulk-max-size bermigrasi menggunakan tabel sumber atau tampilan pencarian, mirip dengan mode LOB lengkap standar. Jika tidak, LOB hingga dan termasuk ukuran ini bermigrasi inline, mirip dengan modus LOB terbatas. Tidak ada LOB yang pernah terpotong diunlimited mode, terlepas dari bentuk yang Anda gunakan.</p> <ul style="list-style-type: none"> • none— Semua tabel atau tampilan LOB dimigrasi sesuai dengan pengaturan LOB tugas. <p>Untuk informasi lebih lanjut tentang pengaturan kebijakan cache, lihat Menargetkan pengaturan tugas metadata.</p> <p>Untuk informasi selengkapnya tentang cara memigrasi LOB dan cara menentukan pengaturan LOB tugas ini, lihat Mengatur dukungan LOB untuk database sumber dalam suatu tugas AWS DMS.</p>	

Parameter	Kemungkinan nilai	Deskripsi
<code>bulk-max-size</code>	Efek dari nilai ini tergantung padamode.	Ukuran maksimum LOB dalam kelipatan kilobyte. Tentukan opsi ini hanya jika Anda perlu mereplika si LOB kecil atau jika target endpoint tidak mendukung ukuran LOB tak terbatas.

Wildcard dalam pengaturan tabel dibatasi

Menggunakan persen wildcard ("%") dalam "table-settings" aturan tidak didukung untuk database sumber seperti yang ditunjukkan berikut.

```
{
  "rule-type": "table-settings",
  "rule-id": "8",
  "rule-name": "8",
  "object-locator": {
    "schema-name": "ipeline-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
  }
}
```

Jika Anda menggunakan "%" dalam "table-settings" aturan seperti yang ditunjukkan, kemudian AWS DMS mengembalikan pengecualian berikut.

```
Error in mapping rules. Rule with ruleId = x failed validation. Exact
schema and table name required when using table settings rule.
```

Selain itu, AWS merekomendasikan agar Anda tidak memuat banyak koleksi besar menggunakan satu tugas dengan `parallel-load`. Perhatikan bahwa AWS DMS membatasi pertentangan sumber daya serta jumlah segmen yang dimuat secara paralel dengan nilai parameter pengaturan `MaxFullLoadSubTasks` tugas, dengan nilai maksimum 49.

Sebagai gantinya, tentukan semua koleksi untuk database sumber Anda untuk koleksi terbesar dengan menentukan masing-masing "schema-name" dan satu "table-name" per satu. Juga, tingkatkan migrasi Anda dengan benar. Misalnya, jalankan beberapa tugas di sejumlah contoh replikasi yang cukup untuk menangani sejumlah besar koleksi besar dalam database Anda.

Menggunakan beban paralel untuk tabel, tampilan, dan koleksi yang dipilih

Untuk mempercepat migrasi dan membuatnya lebih efisien, Anda dapat menggunakan beban paralel untuk tabel relasional, tampilan, dan koleksi yang dipilih. Dengan kata lain, Anda dapat memigrasikan satu tabel, tampilan, atau koleksi tersegmentasi menggunakan beberapa utas secara paralel. Untuk melakukan ini, AWS DMS pisahkan tugas beban penuh menjadi utas, dengan setiap segmen tabel dialokasikan ke utasnya sendiri.

Dengan menggunakan proses pemuatan paralel ini, pertama-tama Anda dapat meminta beberapa utas membongkar beberapa tabel, tampilan, dan koleksi secara paralel dari titik akhir sumber. Anda kemudian dapat memiliki beberapa thread yang bermigrasi dan memuat tabel, tampilan, dan koleksi yang sama secara paralel dengan titik akhir target. Untuk beberapa mesin database, Anda dapat segmen tabel dan pandangan oleh partisi yang ada atau subpartisi. Untuk mesin database lainnya, Anda dapat AWS DMS secara otomatis mengelompokkan koleksi sesuai dengan parameter tertentu (autosegmentasi). Jika tidak, Anda dapat mengelompokkan tabel, tampilan, atau koleksi apa pun berdasarkan rentang nilai kolom yang Anda tentukan.

Beban paralel didukung untuk endpoint sumber berikut:

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2
- SAP Adaptive Server Enterprise (ASE)
- MongoDB (hanya mendukung opsi autosegmentasi dan segmentasi rentang dari beban penuh paralel)

- Amazon DocumentDB (hanya mendukung opsi segmentasi otomatis dan segmentasi rentang beban penuh paralel)
- Amazon DynamoDB

Untuk titik akhir MongoDB dan Amazon DocumentDB AWS DMS , mendukung tipe data berikut untuk kolom yang merupakan kunci partisi untuk opsi segmentasi rentang beban penuh paralel.

- Dua Kali
- String
- ObjectId
- 32 bit integer
- 64 bit integer

Beban paralel untuk digunakan dengan pengaturan tabel aturan yang didukung untuk target akhir berikut:

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- Amazon S3
- SAP Adaptive Server Enterprise (ASE)
- Amazon Redshift
- MongoDB (hanya mendukung opsi autosegmentasi dan segmentasi rentang dari beban penuh paralel)
- Amazon DocumentDB (hanya mendukung opsi segmentasi otomatis dan segmentasi rentang beban penuh paralel)

Untuk menentukan jumlah maksimum tabel dan tampilan untuk memuat secara paralel, gunakan `MaxFullLoadSubTask` pengaturan tugas.

Untuk menentukan jumlah maksimum utas per tabel atau tampilan untuk target yang didukung dari tugas beban paralel, tentukan lebih banyak segmen menggunakan batas nilai kolom.

⚠ Important

`MaxFullLoadSubTasks` mengontrol jumlah tabel atau segmen tabel untuk memuat secara paralel. `ParallelLoadThreads` mengontrol jumlah thread yang digunakan oleh tugas migrasi untuk mengeksekusi beban secara paralel. Pengaturan ini bersifat multiplikatif. Dengan demikian, jumlah total utas yang digunakan selama tugas pemuatan penuh kira-kira merupakan hasil dari nilai **`ParallelLoadThreads`** dikalikan dengan nilai **`MaxFullLoadSubTasks`** (**`ParallelLoadThreads`** * `MaxFullLoadSubTasks`)
Jika Anda membuat tugas dengan jumlah sub tugas Full Load yang tinggi dan sejumlah besar thread beban paralel, tugas Anda dapat menghabiskan terlalu banyak memori dan gagal.

Untuk menentukan jumlah maksimum thread per tabel untuk Amazon DynamoDB, Amazon Kinesis Data Streams, Apache Kafka, atau Amazon Elasticsearch Service target, gunakan setelan tugas metadata target. `ParallelLoadThreads`

Untuk menentukan ukuran buffer untuk tugas beban paralel saat `ParallelLoadThreads` digunakan, gunakan pengaturan tugas metadata `ParallelLoadBufferSize` target.

Ketersediaan dan pengaturan `ParallelLoadThreads` dan `ParallelLoadBufferSize` tergantung pada titik akhir target.

Untuk informasi selengkapnya tentang pengaturan `ParallelLoadThreads` dan `ParallelLoadBufferSize`, lihat [Menargetkan pengaturan tugas metadata](#). Untuk informasi selengkapnya tentang pengaturan `MaxFullLoadSubTasks`, lihat [Pengaturan tugas beban penuh](#). Untuk informasi yang spesifik untuk menargetkan titik akhir, lihat topik terkait.

Untuk menggunakan beban paralel, membuat aturan pemetaan tabel jenis `table-settings` dengan `parallel-load` Pilihan. Dalam `table-settings` aturan, Anda dapat menentukan kriteria segmentasi untuk satu tabel, tampilan, atau koleksi yang ingin Anda muat secara paralel. Untuk melakukannya, tetapkan parameter `type` dari opsi `parallel-load` untuk salah satu dari beberapa opsi.

Cara melakukannya tergantung pada bagaimana Anda ingin mengelompokkan tabel, tampilan, atau koleksi untuk beban paralel:

- Dengan partisi (atau segmen) - Muat semua tabel yang ada atau lihat partisi (atau segmen) menggunakan tipe. `partitions-auto` Atau muat hanya partisi yang dipilih menggunakan `partitions-list` tipe dengan array partisi tertentu.

Hanya untuk titik akhir MongoDB dan Amazon DocumentDB, muat semua atau koleksi tertentu berdasarkan segmen AWS DMS yang secara otomatis menghitung juga menggunakan jenis dan parameter opsional tambahan. `partitions-auto table-settings`

- (Oracle endpoint saja) Dengan subpartisi — Muat semua tabel yang ada atau lihat subpartisi menggunakan `subpartitions-auto` Jenis. Atau memuat hanya subpartisi yang dipilih menggunakan `partitions-list` etiket dengan dispesifikasikan `subpartitionsArray`.
- Berdasarkan segmen yang Anda tentukan — Muat tabel, tampilan, atau segmen koleksi yang Anda tentukan dengan menggunakan batas nilai kolom. Untuk melakukannya, gunakan `rangestipe` dengan dispesifikasikan `columnsdanboundaries` Susunan.

Note

PostgreSQL endpoint hanya mendukung jenis beban paralel ini. MongoDB dan Amazon DocumentDB sebagai titik akhir sumber mendukung jenis segmentasi rentang ini dan tipe autosegmentasi dari beban penuh paralel (`partitions-auto`).

Untuk mengidentifikasi tabel, tampilan, atau koleksi tambahan untuk dimuat secara paralel, tentukan `table-settings` objek tambahan dengan `parallel-load` opsi.

Dalam prosedur berikut, Anda dapat mengetahui cara mengkodekan JSON untuk setiap jenis beban paralel, dari yang paling sederhana hingga yang paling kompleks.

Untuk menentukan semua tabel, tampilan, atau koleksi partisi, atau semua tabel atau tampilan subpartisi

- Tentukan `parallel-load` dengan baik `partitions-auto` Jenis atau `subpartitions-auto` jenis (tetapi tidak keduanya).

Setiap tabel, tampilan, atau partisi koleksi (atau segmen) atau subpartisi kemudian secara otomatis dialokasikan ke utasnya sendiri.

Untuk beberapa titik akhir, beban paralel mencakup partisi atau subpartisi hanya jika sudah ditentukan untuk tabel atau tampilan. Untuk titik akhir sumber MongoDB dan Amazon

DocumentDB, Anda dapat AWS DMS menghitung partisi (atau segmen) secara otomatis berdasarkan parameter tambahan opsional. Ini termasuk `number-of-partitions`, `collection-count-from-metadata`, `max-records-skip-per-page`, dan `batch-size`.

Untuk menentukan tabel yang dipilih atau tampilan partisi, subpartisi, atau keduanya

1. Tentukan `parallel-load` dengan `partitions-list` jenis.
2. (Opsional) Sertakan partisi dengan menentukan sebuah array dari nama partisi sebagai nilai dari `partitions`.

Setiap partisi yang ditentukan kemudian dialokasikan ke benangnya sendiri.

Important

Untuk endpoint Oracle, pastikan partisi dan subpartisi tidak tumpang tindih saat memilihnya untuk beban paralel. Jika Anda menggunakan partisi dan subpartisi yang tumpang tindih untuk memuat data secara paralel, itu menduplikasi entri, atau gagal karena pelanggaran duplikat kunci utama.

3. (Opsional), Untuk endpoint Oracle saja, sertakan subpartisi dengan menentukan array nama subpartisi sebagai nilai `subpartitions`

Setiap subpartisi yang ditentukan kemudian dialokasikan ke benangnya sendiri.

Note

Beban paralel mencakup partisi atau subpartisi hanya jika sudah ditentukan untuk tabel atau tampilan.

Anda dapat menentukan tabel atau tampilan segmen sebagai rentang nilai kolom. Ketika Anda melakukannya, menyadari karakteristik kolom ini:

- Menentukan kolom diindeks secara signifikan meningkatkan kinerja.
- Anda dapat menentukan hingga 10 kolom.
- Anda tidak dapat menggunakan kolom untuk menentukan batas segmen dengan tipe AWS DMS data berikut: `DOUBLE`, `FLOAT`, `BLOB`, `CLOB`, dan `NCLOB`

- Catatan dengan nilai null tidak direplikasi.

Untuk menentukan tabel, tampilan, atau segmen koleksi sebagai rentang nilai kolom

1. Tentukan `parallel-load` dengan `ranges` Jenis.
2. Mendefinisikan batas antara tabel atau tampilan segmen dengan menentukan array nama kolom sebagai nilai `columns`. Lakukan ini untuk setiap kolom yang Anda ingin menentukan batas antara tabel atau tampilan segmen.

Urutan kolom adalah signifikan. Kolom pertama adalah yang paling signifikan dan kolom terakhir paling tidak signifikan dalam mendefinisikan setiap batas, seperti yang dijelaskan berikut.

3. Mendefinisikan rentang data untuk semua tabel atau tampilan segmen dengan menentukan array batas sebagai nilai `boundaries`. A batas array adalah array dari array kolom-nilai. Untuk melakukannya, ikuti langkah-langkah berikut ini:
 - a. Tentukan setiap elemen dari array kolom-nilai sebagai nilai yang sesuai dengan setiap kolom. A array kolom-nilai mewakili batas atas setiap tabel atau tampilan segmen yang ingin Anda definisikan. Tentukan setiap kolom dalam urutan yang sama yang Anda tentukan kolom `columnsArray`.

Masukkan nilai untuk kolom DATE dalam format yang didukung oleh sumber.

- b. Tentukan setiap larik nilai kolom sebagai batas atas, secara berurutan, dari setiap segmen dari bawah ke `next-to-top` segmen tabel atau tampilan. Jika ada baris di atas batas atas yang Anda tentukan, baris ini menyelesaikan segmen atas tabel atau tampilan. Dengan demikian, jumlah segmen berbasis rentang berpotensi satu lebih dari jumlah batas segmen dalam array batas. Setiap segmen berbasis rentang tersebut dialokasikan ke benangnya sendiri.

Semua data non-null direplikasi, bahkan jika Anda tidak menentukan rentang data untuk semua kolom dalam tabel atau tampilan.

Misalnya, Anda mendefinisikan tiga array nilai kolom untuk kolom COL1, COL2, dan COL3 sebagai berikut.

COL1	COL2	COL3
10	30	105
20	20	120
100	12	99

Anda telah menetapkan batas tiga segmen untuk kemungkinan total empat segmen.

Untuk mengidentifikasi rentang baris untuk mereplikasi untuk setiap segmen, contoh replikasi berlaku pencarian untuk tiga kolom ini untuk masing-masing empat segmen. Pencarian adalah sebagai berikut:

Segmen 1

Mereplikasi semua baris di mana berikut ini benar: Nilai dua kolom pertama kurang dari atau sama dengan mereka yang sesuaiSegmen 1 nilai batas atas. Juga, nilai-nilai kolom ketiga kurang dari yangSegmen 1 nilai batas atas.

Segmen 2

Mereplikasi semua baris (kecualiSegmen 1 baris) di mana berikut ini benar: Nilai dua kolom pertama kurang dari atau sama dengan mereka yang sesuaiSegmen 2 nilai batas atas. Juga, nilai-nilai kolom ketiga kurang dari yangSegmen 2 nilai batas atas.

Segmen 3

Mereplikasi semua baris (kecualiSegmen 2 baris) di mana berikut ini benar: Nilai dua kolom pertama kurang dari atau sama dengan mereka yang sesuaiSegmen 3 nilai batas atas. Juga, nilai-nilai kolom ketiga kurang dari yangSegmen 3 nilai batas atas.

Segmen 4

Mereplikasi semua baris yang tersisa (kecualiSegmen 1, 2, dan 3 baris).

Dalam kasus ini, contoh replikasi menciptakanWHERE klausul untuk memuat setiap segmen sebagai berikut:

Segmen 1

```
((COL1 < 10) OR ((COL1 = 10) AND (COL2 < 30)) OR ((COL1 = 10) AND (COL2 = 30) AND (COL3 < 105)))
```

Segmen 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)))
```

Segmen 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)))
```

Segmen 4

```
NOT ((COL1 < 100) OR ((COL1 = 100) AND (COL2 < 12)) OR ((COL1 = 100) AND (COL2 = 12) AND (COL3 < 99)))
```

Menentukan pengaturan LOB untuk tabel yang dipilih atau tampilan

Anda dapat mengatur pengaturan LOB tugas untuk satu atau lebih tabel dengan menciptakan aturan pemetaan tabel jenis `table-settingsdenganlob-settings` pilihan untuk satu atau lebih `table-settings` objek.

Menentukan pengaturan LOB untuk dipilih tabel atau tampilan didukung untuk endpoint sumber berikut:

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2, tergantung pada `modedanbulk-max-size`, dijelaskan sebagai berikut
- SAP Adaptive Server Enterprise (ASE), tergantung pada `modedanbulk-max-size` pengaturan, seperti yang dijelaskan berikut

Menentukan pengaturan LOB untuk dipilih tabel atau tampilan didukung untuk target akhir berikut:

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- SAP ASE, tergantung pada `modedanbulk-max-size` pengaturan, seperti yang dijelaskan berikut

 Note

Anda dapat menggunakan LOB tipe data hanya dengan tabel dan pandangan yang mencakup kunci primer.

Untuk menggunakan pengaturan LOB untuk tabel yang dipilih atau tampilan, Anda membuat aturan pemetaan tabel jenis `table-settings` dengan `lob-settings` Pilihan. Melakukan hal ini menentukan LOB penanganan untuk tabel atau tampilan diidentifikasi oleh `object-locator` Pilihan. Dalam `table-settings` aturan, Anda dapat menentukan `lob-settings` objek dengan parameter berikut:

- `mode`— Menentukan mekanisme penanganan migrasi LOB untuk tabel yang dipilih atau lihat sebagai berikut:
 - `limited`— Mode LOB terbatas default adalah mode tercepat dan paling efisien. Gunakan mode ini hanya jika semua LOB Anda kecil (dalam ukuran 100 MB) atau titik akhir target tidak mendukung ukuran LOB tak terbatas. Juga jika Anda menggunakan `limited`, semua LOB harus dalam ukuran yang Anda tetapkan untuk `bulk-max-size`.

Dalam mode ini untuk tugas beban penuh, contoh replikasi bermigrasi semua LOB inline bersama-sama dengan jenis data kolom lainnya sebagai bagian dari tabel utama atau tampilan penyimpanan. Namun, contoh memotong LOB bermigrasi lebih besar dari `bulk-max-size` nilai untuk ukuran yang ditentukan. Untuk perubahan data capture (CDC) beban tugas, contoh bermigrasi semua LOB menggunakan pencarian tabel sumber, seperti dalam modus LOB penuh standar (lihat berikut). Hal itu tidak terlepas dari ukuran LOB.

 Note

Anda dapat memigrasi tampilan untuk tugas penuh saja.

- `unlimited`— Mekanisme migrasi untuk mode LOB penuh tergantung pada nilai yang Anda tetapkan untuk `bulk-max-size` sebagai berikut:
 - Mode LOB penuh standar— Saat Anda mengatur `bulk-max-size` ke nol, contoh replikasi bermigrasi semua LOB menggunakan standar modus LOB penuh. Mode ini memerlukan pencarian dalam tabel sumber atau tampilan untuk bermigrasi setiap LOB, terlepas dari ukuran. Pendekatan ini biasanya menghasilkan migrasi jauh lebih lambat daripada untuk modus LOB terbatas. Gunakan mode ini hanya jika semua atau sebagian besar LOB Anda besar (1 GB atau lebih besar).
 - Kombinasi mode LOB penuh— Saat Anda mengatur `bulk-max-size` ke nilai nol, mode LOB penuh ini menggunakan kombinasi mode LOB terbatas dan standar modus LOB penuh. Itu adalah untuk tugas beban penuh, jika ukuran LOB dalam `bulk-max-size` nilai, misalnya bermigrasi LOB inline seperti dalam modus LOB terbatas. Jika ukuran LOB lebih besar dari nilai ini, contoh bermigrasi LOB menggunakan tabel sumber atau tampilan pencarian seperti dalam mode LOB penuh standar. Untuk perubahan data capture (CDC) beban tugas, contoh bermigrasi semua LOB menggunakan pencarian tabel sumber, seperti dalam modus LOB penuh standar (lihat berikut). Hal itu tidak terlepas dari ukuran LOB.

 Note

Anda dapat memigrasi tampilan untuk tugas penuh saja.

Mode ini menghasilkan kecepatan migrasi yang kompromi antara lebih cepat, modus LOB terbatas dan lambat, modus LOB penuh standar. Gunakan mode ini hanya jika Anda memiliki campuran LOB kecil dan besar, dan sebagian besar LOB kecil.

Kombinasi mode LOB penuh ini tersedia hanya untuk endpoint berikut:

- IBM Db2 sebagai sumber
- SAP ASE sebagai sumber atau target

Terlepas dari mekanisme yang Anda tentukan untuk mode `unlimited`, contoh ini memigrasikan semua LOB sepenuhnya, tanpa pemotongan.

- `none`— Contoh replikasi memigrasikan LOB dalam tabel yang dipilih atau tampilan menggunakan pengaturan LOB tugas Anda. Gunakan opsi ini untuk membantu membandingkan hasil migrasi dengan dan tanpa pengaturan LOB untuk tabel atau tampilan yang dipilih.

Jika tabel atau tampilan tertentu memiliki LOB yang ada di dalam replikasi, Anda dapat mengatur pengaturan tugas `BatchApplyEnabled` ke `true` hanya ketika menggunakan Mode LOB `limited`.

Dalam beberapa kasus, Anda mungkin mengatur `BatchApplyEnabled` ke `true` dan `BatchApplyPreserveTransaction` ke `false`. Dalam kasus ini, contoh mengatur `BatchApplyPreserveTransaction` ke `true` jika tabel atau tampilan memiliki LOB dan sumber dan target titik akhir adalah Oracle.

- `bulk-max-size`— Set nilai ini ke nilai nol atau bukan nol dalam kilobyte, tergantung pada mode seperti yang dijelaskan untuk item sebelumnya. Masuk `limited` mode, Anda harus menetapkan nilai bukan nol untuk parameter ini.

Contoh mengkonversi LOB ke format biner. Oleh karena itu, untuk menentukan LOB terbesar Anda perlu mereplikasi, kalikan ukurannya dengan tiga. Misalnya, jika LOB terbesar Anda adalah 2 MB, mengatur `bulk-max-size` hingga 6.000 (6 MB).

Contoh pengaturan tabel

Berikut ini, Anda dapat menemukan beberapa contoh yang menunjukkan penggunaan pengaturan tabel.

Example Memuat tabel tersegmentasi oleh partisi

Contoh berikut memuat SALES tabel di sumber Anda lebih efisien dengan memuatnya secara paralel berdasarkan semua partisi.

```
{
  "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 Memuat tabel tersegmentasi oleh subpartisi

Contoh berikut memuat SALES tabel di sumber Oracle Anda lebih efisien dengan memuat secara paralel berdasarkan semua subpartisi nya.

```

{
  "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 Memuat tabel tersegmentasi oleh daftar partisi

Contoh berikut memuat SALES tabel di sumber Anda dengan memuatnya secara paralel dengan daftar partisi tertentu. Di sini, partisi yang ditentukan diberi nama setelah nilai dimulai dengan bagian dari alfabet, misalnya ABCD, EFGH, dan sebagainya.

```

{
  "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 Memuat tabel Oracle tersegmentasi oleh daftar yang dipilih dari partisi dan subpartisi

Contoh berikut memuatSALEStabel di sumber Oracle Anda dengan memuatnya secara paralel dengan daftar partisi dan subpartisi yang dipilih. Di sini, partisi yang ditentukan diberi nama setelah nilai dimulai dengan bagian dari alfabet, misalnyaABCD,EFGH, dan sebagainya. Subpartisi yang ditentukan diberi nama sesuai nilai yang dimulai dengan angka, misalnya01234dan56789.

```
{
  "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 Memuat tabel tersegmentasi dengan rentang nilai kolom

Contoh berikut memuatSALEStabel di sumber Anda dengan memuatnya secara paralel oleh segmen yang ditentukan oleh rentangSALES_NOdanREGIONnilai kolom.

```

{
  "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"
        ]
      ]
    }
  }
]
}

```

```

    ]
  }
}

```

Di sini, dua kolom yang ditentukan untuk rentang segmen dengan nama, SALES_NO dan REGION. Dua batas ditentukan dengan dua set nilai kolom (["1000", "NORTH"] dan ["3000", "WEST"]).

Kedua batas tersebut mengidentifikasi tiga segmen tabel berikut untuk memuat secara paralel:

Segmen 1

Baris dengan SALES_NO kurang dari atau sama dengan 1.000 dan REGION kurang dari "UTARA". Dengan kata lain, angka penjualan hingga 1.000 di wilayah TIMUR.

Segmen 2

Baris selain dari Segmen 1 dengan SALES_NO kurang dari atau sama dengan 3.000 dan REGION kurang dari "BARAT". Dengan kata lain, jumlah penjualan lebih dari 1.000 hingga 3.000 di wilayah UTARA dan SELATAN.

Segmen 3

Semua baris yang tersisa selain Segmen 1 dan Segmen 2. Dengan kata lain, jumlah penjualan lebih dari 3.000 di wilayah "WEST".

Example Memuat dua tabel: Satu tersegmentasi berdasarkan rentang dan lain tersegmentasi oleh partisi

Contoh berikut memuat SALES tabel secara paralel dengan batas-batas segmen yang Anda mengidentifikasi. Hal ini juga memuat ORDER tabel secara paralel oleh semua partisi, seperti contoh sebelumnya.

```

{
  "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 Memuat meja dengan LOB menggunakan modus LOB terbatas

Contoh berikut memuat ITEMStabel termasuk LOB di sumber Anda menggunakan modus LOB terbatas (default) dengan maksimum nontruncated ukuran 100 MB. Setiap LOB yang lebih besar dari ukuran ini dipotong menjadi 100 MB. Semua LOB dimuat inline dengan semua jenis data kolom lainnya.

```
{
  "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 Memuat meja dengan LOB menggunakan modus LOB penuh standar

Contoh berikut memuat tabel ITEMS di sumber Anda, termasuk semua LOB tanpa pemotongan, menggunakan modus LOB penuh standar. Semua LOB, terlepas dari ukuran, dimuat secara terpisah dari jenis data lain menggunakan lookup untuk setiap LOB dalam tabel sumber.

```
{
  "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 Memuat meja dengan LOB menggunakan kombinasi modus LOB penuh

Contoh berikut memuat ITEMStabel di sumber Anda, termasuk semua LOB tanpa pemotongan, menggunakan kombinasi modus LOB penuh. Semua LOB dalam 100 MB dalam ukuran dimuat inline bersama dengan jenis data lainnya, seperti dalam modus LOB terbatas. Semua LOB lebih dari 100 MB dalam ukuran dimuat secara terpisah dari jenis data lainnya. Beban terpisah ini menggunakan pencarian untuk masing-masing LOB tersebut dalam tabel sumber, seperti dalam modus LOB penuh standar.

```

{
  "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 Memuat meja dengan LOB menggunakan pengaturan LOB tugas

Contoh berikut memuat ITEMStabel di sumber Anda, termasuk semua LOB, menggunakan pengaturan LOB tugasnya. Parameter `bulk-max-size` pengaturan 100 MB diabaikan dan dibiarkan hanya untuk reset cepat `limited` atau `unlimited` Mode.

```

{
  "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"
    }
}
]
```

Menggunakan filter sumber

Anda dapat menggunakan filter sumber untuk membatasi jumlah dan jenis catatan yang ditransfer dari sumber ke target. Misalnya, Anda dapat menentukan bahwa hanya karyawan di markas yang dipindahkan ke basis data target. Filter adalah bagian dari aturan seleksi. Terapkan filter pada kolom data.

Filter sumber harus mengikuti kendala ini:

- Aturan seleksi tidak dapat memiliki filter atau satu atau lebih filter.
- Setiap filter dapat memiliki satu atau beberapa kondisi filter.
- Jika lebih dari satu filter digunakan, daftar filter digabungkan seolah-olah menggunakan operator AND antara filter.
- Jika lebih dari satu kondisi filter digunakan dalam filter tunggal, daftar kondisi filter digabungkan seolah-olah menggunakan operator OR antara kondisi filter.
- Filter hanya diterapkan bila `rule-action = 'include'`.
- Filter memerlukan nama kolom dan daftar kondisi filter. Kondisi filter harus memiliki operator filter yang terkait dengan salah satu nilai, dua nilai, atau tidak ada nilai, tergantung pada operator.
- nama kolom, nama tabel, nama tampilan, dan nama skema case-sensitive. Oracle dan Db2 harus selalu menggunakan UPPER case.
- Filter hanya mendukung tabel dengan nama yang tepat. Filter tidak mendukung wildcard.

Batasan berikut berlaku untuk menggunakan filter sumber:

- Filter tidak menghitung kolom right-to-left bahasa.
- Jangan menerapkan filter ke kolom LOB.
- Terapkan filter hanya untukberubahkolom, yang tidak diperbarui setelah pembuatan. Jika filter sumber diterapkan kebisa berubahkolom, yang dapat diperbarui setelah penciptaan, perilaku yang merugikan dapat mengakibatkan.

Misalnya, filter untuk mengecualikan atau menyertakan baris tertentu dalam kolom selalu mengecualikan atau termasuk baris tertentu bahkan jika baris kemudian berubah. Misalkan Anda mengecualikan atau menyertakan baris 1-10 di kolom A, dan mereka kemudian berubah menjadi baris 11-20. Dalam hal ini, mereka terus dikecualikan atau disertakan bahkan ketika data tidak lagi sama.

Demikian pula, anggaplah bahwa baris di luar cakupan filter kemudian diperbarui (atau diperbarui dan dihapus), dan kemudian harus dikecualikan atau disertakan seperti yang didefinisikan oleh filter. Dalam kasus ini, itu direplikasi pada target.

Membuat aturan filter sumber di JSON

Anda dapat membuat filter sumber menggunakan parameter `filters` JSON dari aturan pilihan. Parameter `filters` menentukan array dari satu atau lebih objek JSON. Setiap objek memiliki parameter yang menentukan jenis filter sumber, nama kolom, dan kondisi filter. Kondisi filter ini mencakup satu atau lebih operator filter dan nilai filter.

Tabel berikut menunjukkan parameter untuk menentukan penyaringan sumber dalam `filters` objek.

Parameter	Nilai
<code>filter-type</code>	<code>source</code>
<code>column-name</code>	Parameter dengan nama kolom sumber di mana Anda ingin filter diterapkan. Nama domain tidak peka huruf besar/kecil.
<code>filter-conditions</code>	Array dari satu atau lebih objek yang berisi <code>filter-operator</code> parameter dan nol atau lebih terkait nilai parameter, tergantung pada <code>filter-operator</code> nilai.
<code>filter-operator</code>	Parameter dengan salah satu nilai berikut: <ul style="list-style-type: none"> <code>lte</code>— kurang dari atau sama dengan satu nilai <code>ste</code>— kurang dari atau sama dengan satu nilai (<code>lte</code> Alias) <code>gte</code>— lebih besar dari atau sama dengan satu nilai <code>eq</code>— sama dengan satu nilai <code>noteq</code>— tidak sama dengan satu nilai

Parameter	Nilai
	<ul style="list-style-type: none"> • <code>between</code>— sama dengan atau antara dua nilai • <code>notbetween</code> — tidak sama dengan atau diantara dua nilai • Nilai <code>null</code> – <code>NULL</code> • <code>notnull</code>— tidak<code>NULL</code>nilai
<p><code>value</code> atau</p> <p><code>start-value</code> dan <code>end-value</code> atau</p> <p>tidak ada nilai</p>	<p>Nol atau lebih parameter nilai yang terkait dengan <code>filter-operator</code> :</p> <ul style="list-style-type: none"> • Jika <code>filter-operator</code> adalah <code>lte</code>, <code>ste</code>, <code>gte</code>, <code>eq</code>, atau <code>noteq</code>, gunakan <code>value</code> untuk menentukan satu parameter nilai. • Jika <code>filter-operator</code> adalah <code>between</code> atau <code>notbetween</code> , gunakan <code>start-value</code> dan <code>end-value</code> untuk menentukan dua parameter nilai. • Jika <code>filter-operator</code> adalah <code>null</code> atau <code>notnull</code>, tentukan tidak ada parameter nilai.

Contoh berikut menunjukkan beberapa cara umum untuk menggunakan filter sumber.

Example Filter tunggal

Filter berikut mereplikasi semua karyawan di mana `empid >= 100` ke basis data target.

```
{
  "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 Operator filter multipel

Filter berikut berlaku beberapa operator filter untuk satu kolom data. Filter mereplikasi semua karyawan di mana ($empid \leq 10$) ATAU ($empid$ is between 50 and 75) ATAU ($empid \geq 100$) ke basis data target.

```

{
  "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 Filter beberapa

Filter berikut menerapkan beberapa filter untuk dua kolom dalam sebuah tabel. Filter mereplikasi semua karyawan di mana (`empid <= 100`) DAN (`dept = tech`) ke basis data target.

```
{
  "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 Nilai NULL filter

Filter berikut menunjukkan bagaimana cara untuk memfilter nilai-nilai kosong. Ini mereplikasi semua karyawan di mana `dept = NULL` ke basis data target.

```
{
  "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 Penyaringan menggunakan operator NOT

Beberapa operator dapat digunakan dalam bentuk negatif. Filter berikut mereplikasi semua karyawan di mana (`empid is < 50`) OR (`empid is > 75`) ke basis data target.

```
{
  "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 Menggunakan operator filter Campuran

Mulai dengan AWS DMS versi 3.5.0, Anda dapat mencampur operator inklusif dan operator negatif.

Filter berikut mereplikasi semua karyawan di mana (`empid != 50`) AND (`dept is not NULL`) ke basis data target.

```

{
  "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"
      }]
    }]
  }]
}

```

Perhatikan hal berikut saat menggunakan `null` dengan operator filter lain:

- Menggunakan kondisi inklusif, negatif dan `null` filter bersama-sama dalam filter yang sama tidak akan mereplikasi catatan dengan `NULL` nilai.
- Menggunakan kondisi negatif dan `null` filter bersama-sama tanpa kondisi filter inklusif dalam filter yang sama tidak akan mereplikasi data apa pun.
- Menggunakan kondisi filter negatif tanpa set kondisi `null` filter secara eksplisit tidak akan mereplikasi catatan dengan nilai. `NULL`

Penyaringan berdasarkan waktu dan tanggal

Saat memilih data yang akan diimpor, Anda dapat menentukan tanggal atau waktu sebagai bagian dari kriteria filter. AWS DMS menggunakan format tanggal `YYYY-MM-DD` dan format waktu `YYYY-MM-DD HH:MM:SS` untuk filtering. Parameter AWS DMS fungsi perbandingan mengikuti konvensi SQLite. Untuk informasi lebih lanjut tentang jenis data SQLite dan perbandingan tanggal, lihat [Tipe data dalam SQLite versi 3](#) dalam dokumentasi SQLite.

Filter berikut menunjukkan bagaimana untuk memfilter pada sebuah tanggal. Ini mereplikasi semua karyawan di mana `empstartdate >= January 1, 2002` ke basis data target.

Example Filter tanggal tunggal

```
{
  "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"
      }]
    }]
  }]
}
```

```
    }]  
  }]  
}
```

Mengaktifkan dan bekerja dengan penilaian perdana untuk tugas

Penilaian perdana mengevaluasi komponen tertentu dari tugas migrasi database untuk membantu mengidentifikasi masalah yang mungkin mencegah tugas migrasi berjalan seperti yang diharapkan. Penilaian ini memberi Anda kesempatan untuk mengidentifikasi dan memperbaiki masalah sebelum Anda menjalankan tugas baru atau yang dimodifikasi. Ini memungkinkan Anda untuk menghindari penundaan terkait dengan kegagalan tugas yang disebabkan oleh persyaratan yang hilang atau batasan yang diketahui.

AWS DMS menyediakan akses ke dua opsi berbeda untuk penilaian premi:

- Penilaian tipe data: Laporan lama yang menyediakan lingkup penilaian terbatas.
- Penilaian premigrasi dijalankan: Berisi berbagai jenis penilaian individu, termasuk hasil penilaian tipe data.

Note

Jika Anda memilih menjalankan penilaian premi, Anda tidak perlu memilih penilaian tipe data secara terpisah.

Opsi-opsi ini dijelaskan dalam topik-topik berikut:

- [Menentukan, memulai, dan melihat penilaian perdana berjalan](#): Penilaian premigrasi (disarankan) menentukan satu atau lebih penilaian individu untuk dijalankan berdasarkan konfigurasi tugas migrasi baru atau yang sudah ada. Setiap penilaian individu mengevaluasi elemen tertentu dari basis data sumber dan/atau target yang didukung dari perspektif kriteria seperti jenis migrasi, objek yang didukung, konfigurasi indeks, dan pengaturan tugas lainnya, seperti pemetaan tabel yang mengidentifikasi skema dan tabel untuk bermigrasi.

Misalnya, penilaian individu mungkin mengevaluasi tipe data sumber atau format kunci utama apa yang dapat atau tidak dapat dimigrasikan, mungkin berdasarkan versi AWS DMS mesin. Anda dapat memulai dan melihat hasil proses penilaian terbaru dan melihat hasil dari semua proses penilaian sebelumnya untuk tugas baik menggunakan Konsol AWS DMS Manajemen atau

menggunakan SDK AWS CLI dan untuk mengakses AWS DMS API. Anda juga dapat melihat hasil penilaian sebelumnya yang dijalankan untuk tugas di bucket Amazon S3 yang telah Anda pilih AWS DMS untuk menyimpan hasil ini.

Note

Jumlah dan jenis penilaian individu yang tersedia dapat meningkat dari waktu ke waktu. Untuk informasi lebih lanjut tentang pembaruan OS, lihat [Menentukan penilaian individu](#).

- [Memulai dan melihat penilaian tipe data \(Legacy\)](#) Penilaian tipe data (warisan) mengembalikan hasil dari satu jenis penilaian premi dalam satu struktur JSON: tipe data yang mungkin tidak dimigrasikan dengan benar dalam contoh database sumber relasional yang didukung. Laporan ini menampilkan hasil untuk semua tipe data bermasalah yang ditemukan di setiap skema dan tabel dalam database sumber yang dipilih untuk migrasi.

Membuat prasyarat untuk penilaian premi

Bagian ini menjelaskan sumber daya Amazon S3 dan IAM yang Anda perlukan untuk membuat penilaian premi.

Buat Bucket S3

AWS DMS menyimpan laporan penilaian premi dalam ember S3. Untuk membuat bucket S3, lakukan hal berikut:

1. [Masuk ke AWS Management Console dan buka konsol Amazon S3 di https://console.aws.amazon.com/s3/](https://console.aws.amazon.com/s3/).
2. Pilih Buat bucket.
3. *Pada halaman **Buat bucket**, masukkan nama unik global yang menyertakan nama login untuk bucket, seperti `dms-bucket-yoursignin`.*
4. Pilih Wilayah AWS untuk tugas migrasi DMS.
5. Biarkan pengaturan yang tersisa apa adanya, dan pilih Buat ember.

Buat sumber daya IAM

DMS menggunakan peran dan kebijakan IAM untuk mengakses bucket S3 untuk menyimpan hasil penilaian premi.

Untuk membuat kebijakan IAM, lakukan hal berikut:

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat kebijakan.
4. Di halaman Buat kebijakan, pilih tab JSON.
5. Tempelkan kode JSON berikut ke editor, ganti kode contoh. Ganti *my-bucket* dengan nama bucket Amazon S3 yang Anda buat di bagian sebelumnya.

```
{
  "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. Pilih Berikutnya: Tag, lalu pilih dan Berikutnya: Tinjau.
7. Masukkan **DMSPremigrationAssessmentS3Policy** Nama*, lalu pilih Buat kebijakan.

Untuk membuat peran IAM, lakukan hal berikut:

1. Di konsol IAM, di panel navigasi, pilih Peran.
2. Pilih Buat peran.
3. Pada halaman Pilih entitas tepercaya, untuk jenis entitas tepercaya, pilih AWS Layanan. Untuk kasus penggunaan untuk AWS layanan lain, pilih DMS.
4. Centang kotak centang DMS, lalu pilih Berikutnya.
5. Pada halaman Tambahkan izin, pilih DMS PremigrationAssessment S3Policy. Pilih Berikutnya.
6. Pada halaman Nama, tinjau, dan buat, masukkan **DMSPremigrationAssessmentS3Role** nama Peran, lalu pilih Buat peran.
7. Pada halaman Peran, masukkan nama **DMSPremigrationAssessmentS3Role** Peran. Pilih DMS PremigrationAssessment S3Role.
8. Pada halaman DMS PremigrationAssessment S3Role, pilih tab Trust relationship. Pilih Edit kebijakan kepercayaan.
9. Pada halaman Edit kebijakan kepercayaan, tempelkan JSON berikut ke editor, ganti teks yang ada.

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

Kebijakan ini memberikan `sts:AssumeRole` izin kepada DMS untuk memasukkan hasil penilaian premi ke dalam bucket S3.

10. Pilih Perbarui kebijakan.

Menentukan, memulai, dan melihat penilaian perdana berjalan

Penilaian premi menentukan satu atau lebih penilaian individu untuk dijalankan berdasarkan konfigurasi tugas migrasi baru atau yang sudah ada. Setiap penilaian individu mengevaluasi elemen tertentu dari sumber atau target database tergantung pada pertimbangan seperti jenis migrasi, objek yang didukung, konfigurasi indeks, dan pengaturan tugas lainnya, seperti pemetaan tabel untuk mengidentifikasi skema dan tabel untuk bermigrasi. Misalnya, penilaian individu mungkin mengevaluasi tipe data sumber atau format kunci primer apa yang dapat dan tidak dapat dimigrasikan.

Menentukan penilaian individu

Saat membuat proses penilaian baru, Anda dapat memilih untuk menjalankan beberapa atau semua penilaian individual yang berlaku untuk konfigurasi tugas Anda.

AWS DMS mendukung penilaian premi berjalan untuk sumber relasional berikut dan mesin database target:

- [Penilaian Oracle](#)
- [Penilaian Sql Server](#)
- [Penilaian MySQL](#) (termasuk MariaDB dan Amazon Aurora Edisi yang kompatibel dengan MySQL)
- [Penilaian PostgreSQL](#) (termasuk Amazon Aurora PostgreSQL Edisi yang kompatibel)

Memulai dan melihat penilaian premigrasi berjalan

Anda dapat memulai penilaian premi yang dijalankan untuk tugas migrasi baru atau yang sudah ada menggunakan Konsol AWS DMS Manajemen, Konsol AWS CLI, dan API. AWS DMS

Untuk memulai penilaian perdana untuk tugas baru atau yang sudah ada

1. Dari Tugas migrasi database halaman dalam AWS DMS Console manajemen, lakukan salah satu hal berikut:
 - Untuk membuat tugas baru dan menilainya, pilih **Buat tugas**. Parameter **Buat** halaman tugas migrasi basis data terbuka:
 1. Masukkan pengaturan tugas yang diperlukan untuk membuat tugas Anda, termasuk pemetaan tabel.

2. Di bagian Penilaian Premigrasi, kotak centang Jalankan penilaian Premigrasi dicentang. Halaman ini berisi opsi untuk menentukan penilaian yang dijalankan untuk tugas baru.

 Note

Apabila membuat tugas baru, membolehkan penilaian perdana menjalankan melumpuhkan pilihan untuk memulakan tugas secara automatik pada penciptaan tugas. Anda dapat memulai tugas secara manual setelah menjalankan penilaian selesai.

- Untuk menilai tugas yang ada, pilih Identifier untuk tugas yang ada di halaman tugas migrasi Database. Halaman tugas untuk tugas yang ada dipilih akan terbuka:
 1. PilihTindakandan pilihBuat penilaian perdana. ABuat penilaian perdanamembuka dengan pilihan untuk menentukan penilaian yang dijalankan untuk tugas yang ada.
- 2. Masukkan nama unik untuk menjalankan penilaian Anda, atau tinggalkan nilai default.
- 3. Pilih penilaian individu yang tersedia yang ingin Anda sertakan dalam menjalankan penilaian ini. Anda hanya dapat memilih penilaian individu yang tersedia berdasarkan pengaturan tugas Anda saat ini. Secara default, semua penilaian individu yang tersedia diaktifkan dan dipilih.
- 4. Mencari dan memilih bucket Amazon S3 dan folder di akun Anda untuk menyimpan laporan hasil penilaian Anda. Untuk informasi tentang menyiapkan sumber daya untuk menjalankan penilaian, lihat[Membuat prasyarat untuk penilaian premi](#).
- 5. Pilih atau masukkan IAM role dengan akses akun penuh ke bucket dan folder Amazon S3 yang Anda pilih. Untuk informasi tentang menyiapkan sumber daya untuk menjalankan penilaian, lihat[Membuat prasyarat untuk penilaian premi](#).
- 6. Secara opsional, pilih pengaturan untuk mengenkripsi laporan hasil penilaian di bucket Amazon S3 Anda. Untuk informasi tentang enkripsi bucket S3, lihat [Menyetel perilaku enkripsi sisi server default untuk bucket Amazon S3](#).
- 7. PilihBuat tugasuntuk tugas baru atau pilihBuatuntuk tugas yang ada.

Halaman Tugas migrasi basis data membuka daftar tugas baru atau diubah dengan Status Membuat... dan pesan banner yang menunjukkan bahwa penilaian premigrasi yang dijalankan akan dimulai setelah tugas dibuat.

AWS DMS menyediakan akses ke penilaian premi terbaru dan sebelumnya yang berjalan menggunakan AWS DMS Management Console, the AWS CLI, atau API. AWS DMS

Untuk melihat hasil untuk menjalankan penilaian

1. Dari AWS DMS Management Console, pilih Identifier untuk tugas yang ada di halaman tugas migrasi Database. Halaman tugas untuk tugas yang ada terbuka.
2. Pilih Penilaian Premigrasi tab pada halaman tugas yang ada. Ini membuka bagian penilaian Premigrasi pada halaman tersebut yang menunjukkan hasil penilaian berjalan, yang dicantumkan berdasarkan nama, dalam urutan kronologis terbalik. Hasil terbaru muncul di bagian atas daftar. Pilih nama proses penilaian yang hasilnya ingin Anda lihat.

Hasil lari penilaian ini dimulai dengan nama uji penilaian terbaru dan ikhtisar statusnya diikuti dengan daftar penilaian individu yang ditentukan dan statusnya. Anda kemudian dapat mengeksplorasi rincian status masing-masing penilaian individu dengan memilih namanya dalam daftar, dengan hasil yang tersedia sampai ke tingkat kolom tabel.

Kedua-dua ikhtisar status untuk menjalankan penilaian dan setiap penilaian individu menunjukkan Status nilai. Nilai ini menunjukkan status keseluruhan dari penilaian dan status yang sama untuk setiap penilaian individu. Berikut ini adalah daftar Status nilai-nilai untuk menjalankan penilaian:

- "cancelling"— Penilaian dijalankan dibatalkan.
- "deleting"— Penilaian telah dihapus.
- "failed"— Setidaknya satu penilaian individu dilengkapi dengan failedStatus.
- "error-provisioning"— Kesalahan internal terjadi ketika sumber daya ditetapkan (selama provisioningStatus).
- "error-executing"— Terjadi kesalahan internal saat penilaian individu berjalan (selama runningStatus).
- "invalid state"— Penilaian berjalan dalam keadaan tidak diketahui.
- "passed"— Semua penilaian individu telah selesai, dan tidak ada yang memiliki failedStatus.
- "provisioning"— Sumber daya yang diperlukan untuk menjalankan penilaian individu sedang ditetapkan.
- "running"— Penilaian individu sedang dijalankan.
- "starting"— Penilaian berjalan dimulai, tetapi sumber daya belum ditetapkan untuk penilaian individu.
- "warning"— Setidaknya satu penilaian individu dilengkapi dengan warningStatus.

Berikut ini adalah daftarStatusnilai-nilai untuk setiap penilaian individu menjalankan penilaian:

- "cancelled"— Penilaian individu dibatalkan sebagai bagian dari pembatalan pelaksanaan penilaian.
- "error"— Penilaian individu tidak berhasil diselesaikan.
- "failed"— Penilaian individu berhasil diselesaikan dengan hasil validasi yang gagal: lihat rincian hasil untuk informasi lebih lanjut.
- "invalid state"— Penilaian individu dalam keadaan tidak diketahui.
- "passed"— Penilaian individu dilengkapi dengan hasil validasi yang berhasil.
- "pending"— Penilaian individu sedang menunggu untuk dijalankan.
- "running"— Penilaian individu sedang berjalan.
- "warning"— Penilaian individu berhasil diselesaikan dengan hasil validasi peringatan: lihat rincian hasil untuk informasi lebih lanjut.

Anda juga dapat melihat file JSON untuk hasil penilaian run pada Amazon S3.

Untuk melihat file JSON untuk penilaian berjalan di Amazon S3

1. Dari AWS DMS Management Console, pilih tautan bucket Amazon S3 yang ditampilkan di ikhtisar status proses penilaian. Ini menampilkan daftar folder bucket dan objek Amazon S3 lainnya yang disimpan dalam bucket. Jika hasil disimpan dalam folder bucket, buka folder tersebut.
2. Anda dapat menemukan hasil penilaian Anda menjalankan dalam beberapa file JSON. File `summary.json` berisi hasil keseluruhan dari penilaian yang dijalankan. Berkas yang tersisa masing-masing diberi nama untuk penilaian individu yang ditentukan untuk menjalankan penilaian, seperti `unsupported-data-types-in-source.json`. File-file ini masing-masing berisi hasil untuk penilaian individu yang sesuai dari penilaian yang dipilih dijalankan.

Untuk memulai dan melihat hasil penilaian premi yang dijalankan untuk tugas migrasi yang ada, Anda dapat menjalankan perintah CLI dan operasi API berikut: AWS DMS

- CLI: [describe-applicable-individual-assessments](#), API: [DescribeApplicableIndividualAssessments](#) – Menyediakan daftar penilaian individu yang dapat Anda tentukan untuk menjalankan penilaian premigrasi baru, diberikan satu atau lebih parameter konfigurasi tugas.

- CLI: [start-replication-task-assessment-run](#), API: [StartReplicationTaskAssessmentRun](#) – Memulai penilaian premigrasi baru yang dijalankan untuk satu penilaian individu atau lebih dari tugas migrasi yang ada.
- CLI: [describe-replication-task-assessment-runs](#), API: [DescribeReplicationTaskAssessmentRuns](#) – Mengembalikan daftar yang diberi nomor halaman tentang penilaian premigrasi yang dijalankan berdasarkan pengaturan filter.
- CLI: [describe-replication-task-individual-assessments](#), API: [DescribeReplicationTaskIndividualAssessments](#) – Mengembalikan daftar yang diberi nomor halaman tentang penilaian individu berdasarkan pengaturan filter.
- CLI: [cancel-replication-task-assessment-run](#), API: [CancelReplicationTaskAssessmentRun](#)— Membatalkan, tetapi tidak menghapus, satu penilaian premi berjalan.
- CLI: [delete-replication-task-assessment-run](#), API: [DeleteReplicationTaskAssessmentRun](#)— Menghapus catatan penilaian premigrasi tunggal yang dijalankan.

Penilaian individu

Bagian ini menjelaskan penilaian premi individu.

Untuk membuat penilaian premi individual menggunakan AWS DMS API, gunakan kunci API yang terdaftar untuk IncludeOnly parameter tindakan. [StartReplicationTaskAssessmentRun](#)

Topik

- [Penilaian untuk semua jenis titik akhir](#)
- [Penilaian Oracle](#)
- [Penilaian Sql Server](#)
- [Penilaian MySQL](#)
- [Penilaian MariaDB](#)
- [Penilaian PostgreSQL](#)

Penilaian untuk semua jenis titik akhir

Bagian ini menjelaskan penilaian premi individu untuk semua jenis titik akhir.

Topik

- [Tipe data yang tidak didukung](#)
- [Objek besar \(LOB\) digunakan tetapi kolom LOB target tidak dapat dibatalkan](#)
- [Tabel sumber dengan Objek besar \(LOB\) tetapi tanpa kunci utama atau kendala unik](#)
- [Tabel sumber tanpa kunci utama untuk tugas CDC atau beban penuh dan CDC saja](#)
- [Tabel target tanpa kunci utama hanya untuk tugas CDC](#)
- [Jenis kunci primer sumber yang tidak didukung - kunci primer komposit](#)

Tipe data yang tidak didukung

Kunci API: `unsupported-data-types-in-source`

Memeriksa tipe data di titik akhir sumber yang tidak didukung DMS. Tidak semua tipe data dapat dimigrasi antar mesin.

Objek besar (LOB) digunakan tetapi kolom LOB target tidak dapat dibatalkan

Kunci API: `full-lob-not-nullable-at-target`

Memeriksa nullabilitas kolom LOB di target saat replikasi menggunakan mode LOB penuh atau mode LOB sebaris. DMS membutuhkan kolom LOB menjadi nol saat menggunakan mode LOB ini. Penilaian ini membutuhkan basis data sumber dan target untuk menjadi relasional.

Tabel sumber dengan Objek besar (LOB) tetapi tanpa kunci utama atau kendala unik

Kunci API: `table-with-lob-but-without-primary-key-or-unique-constraint`

Mengecek keberadaan tabel sumber dengan LOB tapi tanpa kunci primer atau khusus. Tabel harus memiliki kunci utama atau kunci unik untuk DMS untuk memigrasikan LOB. Penilaian ini membutuhkan database sumber untuk menjadi relasional.

Tabel sumber tanpa kunci utama untuk tugas CDC atau beban penuh dan CDC saja

Kunci API: `table-with-no-primary-key-or-unique-constraint`

Memeriksa keberadaan kunci primer atau kunci unik dalam tabel sumber untuk migrasi full-load dan change data capture (CDC), atau migrasi khusus CDC. Kurangnya kunci primer atau kunci unik dapat menyebabkan masalah kinerja selama migrasi CDC. Penilaian ini mengharuskan database sumber bersifat relasional, dan jenis migrasi menyertakan CDC.

Tabel target tanpa kunci utama hanya untuk tugas CDC

Kunci API: `target-table-has-unique-key-or-primary-key-for-cdc`

Memeriksa keberadaan kunci primer atau kunci unik dalam tabel target yang sudah dibuat untuk migrasi CDC-saja. Kurangnya kunci utama atau kunci unik dapat menyebabkan pemindaian tabel lengkap di target saat DMS menerapkan pembaruan dan penghapusan. Hal ini dapat mengakibatkan masalah kinerja selama migrasi CDC. Penilaian ini mengharuskan database target bersifat relasional, dan jenis migrasi menyertakan CDC.

Jenis kunci primer sumber yang tidak didukung - kunci primer komposit

Kunci API: `unsupported-source-pk-type-for-elasticsearch-target`

Memeriksa keberadaan kunci utama komposit dalam tabel sumber saat bermigrasi ke Amazon OpenSearch Service. Kunci utama dari tabel sumber harus terdiri dari satu kolom. Penilaian ini membutuhkan database sumber untuk menjadi relasional, dan database target menjadi DynamoDB.

Note

DMS mendukung migrasi database sumber ke target OpenSearch Layanan di mana kunci utama sumber terdiri dari beberapa kolom.

Penilaian Oracle

Bagian ini menjelaskan penilaian premi individu untuk tugas migrasi yang menggunakan titik akhir sumber Oracle.

Note

Untuk menggunakan penilaian premi di bagian ini, Anda harus menambahkan izin berikut ke: `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;  
grant select on dba_users to dms_user;  
grant select on dba_directories to dms_user;
```

Untuk informasi selengkapnya tentang izin saat menggunakan Oracle sebagai sumber, lihat [Hak istimewa akun pengguna diperlukan pada sumber Oracle yang dikelola sendiri untuk AWS DMS](#)

Topik

- [Periksa pencatatan tambahan pada tingkat basis data](#)
- [Validasi jika diperlukan tautan DB dibuat untuk Standby](#)
- [Validasi Oracle untuk tipe data LOB dan jika pembaca biner dikonfigurasi](#)
- [Validasi jika database adalah CDB](#)
- [Periksa Edisi Database Oracle](#)
- [Validasi metode Oracle CDC untuk DMS](#)
- [Validasi konfigurasi Oracle RAC untuk DMS](#)

Periksa pencatatan tambahan pada tingkat basis data

Kunci API: `oracle-supplemental-db-level`

Penilaian premi ini memvalidasi jika logging tambahan minimum diaktifkan di tingkat database. Anda harus mengaktifkan logging tambahan untuk menggunakan database Oracle sebagai sumber migrasi.

Untuk mengaktifkan logging tambahan, gunakan kueri berikut:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA
```

Untuk informasi selengkapnya, lihat [Mengatur supplemental logging](#).

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Validasi jika diperlukan tautan DB dibuat untuk Standby

Kunci API: `oracle-validate-standby-dblink`

Penilaian premi ini memvalidasi jika Dblink dibuat untuk sumber database siaga Oracle. AWS DMS_DBLINK adalah prasyarat untuk menggunakan database siaga sebagai sumber. Saat menggunakan Oracle Standby sebagai sumber, AWS DMS tidak memvalidasi transaksi terbuka secara default.

Untuk informasi selengkapnya, lihat [Bekerja dengan database Oracle yang dikelola sendiri sebagai sumber AWS DMS](#).

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Validasi Oracle untuk tipe data LOB dan jika pembaca biner dikonfigurasi

Kunci API: `oracle-binary-lob-source-validation`

Penilaian premi ini memvalidasi jika Oracle digunakan untuk endpoint database Oracle LogMiner versi 12c atau yang lebih baru. AWS DMS tidak mendukung Oracle LogMiner untuk migrasi kolom LOB dari database Oracle versi 12c. Penilaian ini juga memeriksa keberadaan kolom LOB dan memberikan rekomendasi yang sesuai.

Untuk mengonfigurasi migrasi agar tidak menggunakan Oracle LogMiner, tambahkan konfigurasi berikut ke titik akhir sumber Anda:

```
useLogMinerReader=N;useBfile=Y;
```

Untuk informasi selengkapnya, lihat [Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC](#).

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Validasi jika database adalah CDB

Kunci API: `oracle-validate-cdb`

Penilaian premi ini memvalidasi jika database adalah database kontainer. AWS DMS tidak mendukung database root kontainer multi-penyewa (CDB\$ROOT).

Note

Penilaian ini hanya diperlukan untuk Oracle versi 12.1.0.1 atau yang lebih baru. Penilaian ini tidak berlaku untuk versi Oracle sebelum 12.1.0.1.

Untuk informasi selengkapnya, lihat [Keterbatasan dalam menggunakan Oracle sebagai sumber AWS DMS](#).

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Periksa Edisi Database Oracle

Kunci API: `oracle-check-cdc-support-express-edition`

Penilaian premi ini memvalidasi jika database sumber Oracle adalah Edisi Ekspres. AWS DMS tidak mendukung CDC untuk Oracle Express Edition (Oracle Database XE) versi 18.0 dan yang lebih baru.

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Validasi metode Oracle CDC untuk DMS

Kunci API: `oracle-recommendation-cdc-method`

Penilaian premi ini memvalidasi pembuatan redo log selama tujuh hari terakhir, dan membuat rekomendasi apakah akan menggunakan AWS DMS Binary Reader atau Oracle untuk CDC.

LogMiner

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Untuk informasi selengkapnya tentang memutuskan metode CDC mana yang akan digunakan, lihat [Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC](#).

Validasi konfigurasi Oracle RAC untuk DMS

Kunci API: `oracle-check-rac`

Penilaian premi ini memvalidasi jika database oracle adalah Real Application Cluster. Database Real Application Cluster harus dikonfigurasi dengan benar. Jika database didasarkan pada RAC, kami sarankan Anda menggunakan AWS DMS Binary Reader untuk CDC daripada Oracle. LogMiner

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Untuk informasi selengkapnya, lihat [Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC](#).

Penilaian Sql Server

Bagian ini menjelaskan penilaian premi individu untuk tugas migrasi yang menggunakan titik akhir sumber Microsoft SQL Server.

Topik

- [Periksa apakah model pemulihan untuk database sederhana](#)
- [Periksa apakah tabel dalam lingkup tugas berisi kolom yang dihitung](#)
- [Periksa apakah tabel dalam lingkup tugas memiliki indeks penyimpanan kolom](#)
- [Periksa apakah tabel yang dioptimalkan memori adalah bagian dari cakupan tugas](#)
- [Periksa apakah tabel temporal adalah bagian dari ruang lingkup tugas](#)
- [Periksa apakah daya tahan tertunda diaktifkan di tingkat database](#)
- [Periksa apakah pemulihan data yang dipercepat diaktifkan di tingkat database](#)
- [Periksa apakah pemetaan tabel memiliki lebih dari 10K tabel dengan kunci utama](#)

Periksa apakah model pemulihan untuk database sederhana

Kunci API: `sqlserver-check-for-recovery-model`

Penilaian premi ini memvalidasi model pemulihan titik akhir sumber. AWS DMS mengharuskan model pemulihan disetel ke `Bulk Logged` atau `Full` untuk replikasi yang sedang berlangsung.

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Untuk informasi selengkapnya, lihat [Prasyarat untuk menggunakan replikasi berkelanjutan \(CDC\) dari sumber SQL Server](#).

Periksa apakah tabel dalam lingkup tugas berisi kolom yang dihitung

Kunci API: `sqlserver-check-for-computed-fields`

Penilaian premi ini memeriksa keberadaan kolom yang dihitung. AWS DMS tidak mendukung mereplikasi perubahan dari kolom yang dihitung SQL Server.

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Untuk informasi selengkapnya, lihat [Keterbatasan dalam menggunakan SQL Server sebagai sumber AWS DMS](#).

Periksa apakah tabel dalam lingkup tugas memiliki indeks penyimpanan kolom

Kunci API: `sqlserver-check-for-columnstore-indexes`

Penilaian premi ini memeriksa keberadaan tabel dengan indeks columnstore. AWS DMS tidak mendukung mereplikasi perubahan dari tabel SQL Server dengan indeks columnstore.

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Untuk informasi selengkapnya, lihat [Keterbatasan dalam menggunakan SQL Server sebagai sumber AWS DMS](#).

Periksa apakah tabel yang dioptimalkan memori adalah bagian dari cakupan tugas

Kunci API: `sqlserver-check-for-memory-optimized-tables`

Penilaian premi ini memeriksa keberadaan tabel yang dioptimalkan untuk memori. AWS DMS tidak mendukung replikasi perubahan dari tabel yang dioptimalkan memori.

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Untuk informasi selengkapnya, lihat [Keterbatasan dalam menggunakan SQL Server sebagai sumber AWS DMS](#).

Periksa apakah tabel temporal adalah bagian dari ruang lingkup tugas

Kunci API: `sqlserver-check-for-temporal-tables`

Penilaian premi ini memeriksa keberadaan tabel temporal. AWS DMS tidak mendukung replikasi perubahan dari tabel temporal.

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Untuk informasi selengkapnya, lihat [Keterbatasan dalam menggunakan SQL Server sebagai sumber AWS DMS](#).

Periksa apakah daya tahan tertunda diaktifkan di tingkat database

Kunci API: `sqlserver-check-for-delayed-durability`

Penilaian premi ini memeriksa keberadaan daya tahan yang tertunda. AWS DMS tidak mendukung replikasi perubahan dari transaksi yang menggunakan daya tahan tertunda.

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Untuk informasi selengkapnya, lihat [Keterbatasan dalam menggunakan SQL Server sebagai sumber AWS DMS](#).

Periksa apakah pemulihan data yang dipercepat diaktifkan di tingkat database

Kunci API: `sqlserver-check-for-accelerated-data-recovery`

Penilaian premi ini memeriksa keberadaan pemulihan data yang dipercepat. AWS DMS tidak mendukung replikasi perubahan dari database dengan pemulihan data yang dipercepat.

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Untuk informasi selengkapnya, lihat [Keterbatasan dalam menggunakan SQL Server sebagai sumber AWS DMS](#).

Periksa apakah pemetaan tabel memiliki lebih dari 10K tabel dengan kunci utama

Kunci API: `sqlserver-large-number-of-tables`

Penilaian premi ini memeriksa keberadaan lebih dari 10.000 tabel dengan kunci utama. Database yang dikonfigurasi dengan MS-replikasi dapat mengalami kegagalan tugas jika ada terlalu banyak tabel dengan kunci utama.

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Untuk informasi selengkapnya tentang mengonfigurasi replikasi MS, lihat. [Menangkap perubahan data untuk SQL Server yang dikelola sendiri di tempat atau di Amazon EC2](#)

Penilaian MySQL

Bagian ini menjelaskan penilaian premi individual untuk tugas migrasi yang menggunakan titik akhir sumber MySQL.

Topik

- [Validasi jika tabel menggunakan mesin penyimpanan selain InnoDB](#)
- [Validasi jika kenaikan otomatis diaktifkan pada tabel apa pun yang digunakan untuk migrasi](#)
- [Validasi jika gambar binlog database diatur FULL untuk mendukung DMS CDC](#)
- [Validasi jika database sumber adalah MySQL Read-Replica](#)
- [Validasi jika tabel memiliki partisi, dan merekomendasikan target_table_prep_mode untuk pengaturan tugas beban penuh](#)
- [Validasi jika DMS mendukung versi database](#)
- [Validasi jika database target dikonfigurasi untuk disetel local_infile ke 1](#)
- [Validasi jika database target memiliki tabel dengan kunci asing](#)
- [Validasi jika tabel sumber dalam lingkup tugas memiliki kendala kaskade](#)
- [Validasi jika nilai batas waktu sesuai untuk sumber atau target MySQL](#)

Validasi jika tabel menggunakan mesin penyimpanan selain InnoDB

Kunci API: `mysql-check-table-storage-engine`

Penilaian premi ini memvalidasi apakah mesin penyimpanan yang digunakan untuk tabel apa pun dalam database Sumber MySQL adalah mesin selain InnoDB. DMS membuat tabel target dengan mesin penyimpanan InnoDB secara default. Jika Anda perlu menggunakan mesin penyimpanan selain InnoDB, Anda harus secara manual membuat tabel pada database target dan mengonfigurasi tugas DMS Anda untuk digunakan TRUNCATE_BEFORE_LOAD atau DO_NOTHING sebagai pengaturan tugas beban penuh. Untuk informasi selengkapnya tentang setelan tugas beban penuh, lihat [Pengaturan tugas beban penuh](#).

Untuk informasi selengkapnya tentang batasan titik akhir MySQL, lihat. [Batasan dalam menggunakan database MySQL sebagai sumber AWS DMS](#)

Validasi jika kenaikan otomatis diaktifkan pada tabel apa pun yang digunakan untuk migrasi

Kunci API: `mysql-check-auto-increment`

Penilaian premi ini memvalidasi apakah tabel sumber yang digunakan dalam tugas mengaktifkan penambahan otomatis. DMS tidak memigrasikan atribut AUTO_INCREMENT pada kolom ke database target.

Untuk informasi selengkapnya tentang batasan titik akhir MySQL, lihat. [Batasan dalam menggunakan database MySQL sebagai sumber AWS DMS](#) Untuk informasi tentang penanganan kolom identitas di MySQL, [lihat Menangani kolom IDENTITAS AWS DMS di: Bagian 2.](#)

Validasi jika gambar binlog database diatur **FULL** untuk mendukung DMS CDC

Kunci API: `mysql-check-binlog-image`

Penilaian premi ini memeriksa apakah image binlog database sumber disetel ke. FULL Di MySQL, variabel `binlog_row_image` menentukan bagaimana peristiwa log biner ditulis saat menggunakan format. ROW Untuk memastikan kompatibilitas dengan DMS dan mendukung CDC, atur `binlog_row_image` variabel ke. FULL Pengaturan ini memastikan bahwa DMS menerima informasi yang cukup untuk membangun Data Manipulation Language (DHTML) lengkap untuk database target selama migrasi.

Untuk mengatur gambar binlogFULL, lakukan hal berikut:

- Untuk Amazon RDS, nilai ini secara FULL default.
- Untuk database yang menggunakan selang lokal atau di Amazon EC2, tetapkan nilainya `binlog_row_image` di (`my.ini`Microsoft Windows) atau (`my.cnf` UNIX).

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Validasi jika database sumber adalah MySQL Read-Replica

Kunci API: `mysql-check-database-role`

Penilaian premi ini memverifikasi apakah database sumber adalah replika baca. Untuk mengaktifkan dukungan CDC untuk DMS saat terhubung ke replika baca, atur parameter ke. `log_slave_updates True` Untuk informasi selengkapnya tentang menggunakan database MySQL yang dikelola sendiri, lihat. [Menggunakan database yang kompatibel dengan MySQL yang dikelola sendiri sebagai sumber AWS DMS](#)

Untuk mengatur `log_slave_updates` nilainya `True`, lakukan hal berikut:

- Untuk Amazon RDS, gunakan grup parameter database. Untuk informasi tentang menggunakan grup parameter database RDS, lihat [Bekerja dengan grup parameter](#) di Panduan Pengguna Amazon RDS.
- Untuk database yang menggunakan selang lokal atau di Amazon EC2, tetapkan nilainya `log_slave_updates` di (`my.ini` Microsoft Windows) atau (`my.cnf` UNIX).

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Validasi jika tabel memiliki partisi, dan rekomendasikan **`target_table_prep_mode`** untuk pengaturan tugas beban penuh

Kunci API: `mysql-check-table-partition`

Penilaian premi ini memeriksa keberadaan tabel dengan partisi dalam database sumber. DMS membuat tabel tanpa partisi pada target MySQL. Untuk memigrasikan tabel yang dipartisi ke tabel yang dipartisi pada target, Anda harus melakukan hal berikut:

- Pra-buat tabel yang dipartisi dalam database MySQL target.
- Konfigurasi tugas DMS Anda untuk digunakan `TRUNCATE_BEFORE_LOAD` atau `DO_NOTHING` sebagai pengaturan tugas beban penuh.

Untuk informasi selengkapnya tentang batasan titik akhir MySQL, lihat. [Batasan dalam menggunakan database MySQL sebagai sumber AWS DMS](#)

Validasi jika DMS mendukung versi database

Kunci API: `mysql-check-supported-version`

Penilaian premi ini memverifikasi apakah versi database sumber kompatibel dengan DMS. CDC tidak didukung dengan Amazon RDS MySQL versi 5.5 atau lebih rendah, atau versi MySQL yang lebih besar dari 8.0.x. CDC hanya didukung untuk MySQL versi 5.6, 5.7, atau 8.0. Untuk informasi selengkapnya tentang versi MySQL yang didukung, lihat. [Titik akhir sumber untuk migrasi data](#)

Validasi jika database target dikonfigurasi untuk disetel **`local_infile`** ke 1

Kunci API: `mysql-check-target-localinfile-set`

Penilaian premi ini memeriksa apakah `local_infile` parameter dalam database target diatur ke 1. DMS memerlukan parameter 'local_infile' untuk disetel ke 1 selama pemuatan penuh dalam database target Anda. Untuk informasi selengkapnya, lihat [Migrasi dari MySQL ke MySQL menggunakan AWS DMS](#).

Penilaian ini hanya berlaku untuk tugas beban penuh atau beban penuh dan CDC.

Validasi jika database target memiliki tabel dengan kunci asing

Kunci API: `mysql-check-fk-target`

Penilaian premi ini memeriksa apakah tugas beban penuh atau penuh dan CDC yang bermigrasi ke database MySQL memiliki tabel dengan kunci asing. Pengaturan default dalam DMS adalah memuat tabel dalam urutan abjad. Tabel dengan kunci asing dan batasan integritas referensial dapat menyebabkan beban gagal, karena tabel induk dan anak mungkin tidak dimuat pada saat yang bersamaan.

Untuk informasi selengkapnya tentang integritas referensial di DMS, lihat Bekerja dengan indeks, pemicu, dan kendala integritas referensial dalam topik. [Meningkatkan performa suatu migrasi AWS DMS](#)

Validasi jika tabel sumber dalam lingkup tugas memiliki kendala kaskade

Kunci API: `mysql-check-cascade-constraints`

Penilaian premi ini memeriksa apakah ada tabel sumber MySQL yang memiliki batasan kaskade. Batasan kaskade tidak dimigrasikan atau direplikasi oleh tugas DMS, karena MySQL tidak merekam perubahan untuk peristiwa ini di binlog. Meskipun AWS DMS tidak mendukung kendala ini, Anda dapat menggunakan solusi untuk target database relasional.

Untuk informasi tentang mendukung batasan cascade dan kendala lainnya, lihat [Indeks, Kunci Asing, atau Pembaruan atau Penghapusan Cascade Tidak Dimigrasi](#) di topik tugas migrasi pemecahan masalah. AWS DMS

Validasi jika nilai batas waktu sesuai untuk sumber atau target MySQL

Kunci API: `mysql-check-network-parameter`

Penilaian premi ini memeriksa apakah titik akhir MySQL tugas memilikinet_read_timeout, net_wait_timeout dan wait_timeout pengaturan disetel ke setidaknya 300 detik. Ini diperlukan untuk mencegah pemutusan selama migrasi.

Untuk informasi selengkapnya, lihat [Koneksi ke instans MySQL target terputus selama pengerjaan tugas](#).

Penilaian MariaDB

Bagian ini menjelaskan penilaian premi individu untuk tugas migrasi yang menggunakan titik akhir sumber MariaDB.

Untuk membuat penilaian premi individual menggunakan AWS DMS API, gunakan kunci API yang terdaftar untuk Include parameter tindakan. [StartReplicationTaskAssessmentRun](#)

Topik

- [Validasi jika tabel menggunakan mesin penyimpanan selain Innodb](#)
- [Validasi jika kenaikan otomatis diaktifkan pada tabel apa pun yang digunakan untuk migrasi](#)
- [Validasi jika format binlog database diatur ROW untuk mendukung DMS CDC](#)
- [Validasi jika gambar binlog database diatur FULL untuk mendukung DMS CDC](#)
- [Validasi jika database sumber adalah MariaDB Read-Replica](#)
- [Validasi jika tabel memiliki partisi, dan merekomendasikan TRUNCATE_BEFORE_LOAD atau DO_NOTHING untuk pengaturan tugas beban penuh](#)
- [Validasi jika DMS mendukung versi database](#)
- [Validasi jika database target dikonfigurasi untuk disetel local_infile ke 1](#)
- [Validasi jika database target memiliki tabel dengan kunci asing](#)
- [Validasi jika tabel sumber dalam lingkup tugas memiliki kendala kaskade](#)
- [Validasi jika tabel sumber dalam lingkup tugas telah menghasilkan kolom](#)
- [Validasi jika nilai batas waktu sesuai untuk sumber MariaDB](#)
- [Validasi jika nilai batas waktu sesuai untuk target MariaDB](#)

Validasi jika tabel menggunakan mesin penyimpanan selain Innodb

Kunci API: mariadb-check-table-storage-engine

Penilaian premi ini memvalidasi apakah mesin penyimpanan yang digunakan untuk tabel apa pun di database Sumber MariaDB adalah mesin selain Innodb. DMS membuat tabel target dengan mesin penyimpanan InnoDB secara default. Jika Anda perlu menggunakan mesin penyimpanan selain InnoDB, Anda harus secara manual membuat tabel pada database target dan mengonfigurasi

tugas DMS Anda untuk digunakan TRUNCATE_BEFORE_LOAD atau DO_NOTHING sebagai pengaturan tugas beban penuh. Untuk informasi selengkapnya tentang setelan tugas beban penuh, lihat [Pengaturan tugas beban penuh](#).

Untuk informasi selengkapnya tentang batasan titik akhir MariaDB, lihat. [Batasan dalam menggunakan database MySQL sebagai sumber AWS DMS](#)

Validasi jika kenaikan otomatis diaktifkan pada tabel apa pun yang digunakan untuk migrasi

Kunci API: mariadb-check-auto-increment

Penilaian premi ini memvalidasi apakah tabel sumber yang digunakan dalam tugas mengaktifkan penambahan otomatis. DMS tidak memigrasikan atribut AUTO_INCREMENT pada kolom ke database target.

Untuk informasi selengkapnya tentang batasan titik akhir MariaDB, lihat. [Batasan dalam menggunakan database MySQL sebagai sumber AWS DMS](#) Untuk informasi tentang penanganan kolom identitas di MariaDB, [lihat Menangani kolom IDENTITAS AWS DMS di: Bagian 2](#).

Validasi jika format binlog database diatur **ROW** untuk mendukung DMS CDC

Kunci API: mariadb-check-binlog-format

Penilaian premi ini memvalidasi apakah format binlog basis data sumber diatur ROW untuk mendukung DMS Change Data Capture (CDC).

Untuk mengatur format binlogROW, lakukan hal berikut:

- Untuk Amazon RDS, gunakan grup parameter database. Untuk informasi tentang menggunakan grup parameter RDS, lihat [Mengonfigurasi logging biner MySQL di Panduan Pengguna Amazon RDS](#).
- Untuk database yang menggunakan selang lokal atau di Amazon EC2, tetapkan nilainya binlog_format di (my.iniMicrosoft Windows) atau (UNIX). my.cnf

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Untuk informasi selengkapnya tentang server MariaDB yang dihosting sendiri, lihat. [Menggunakan database yang kompatibel dengan MySQL yang dikelola sendiri sebagai sumber AWS DMS](#)

Validasi jika gambar binlog database diatur **FULL** untuk mendukung DMS CDC

Kunci API: `mariadb-check-binlog-image`

Penilaian premi ini memeriksa apakah image binlog database sumber disetel ke. FULL Di MariaDB, variabel `binlog_row_image` menentukan bagaimana peristiwa log biner ditulis saat menggunakan format. ROW Untuk memastikan kompatibilitas dengan DMS dan mendukung CDC, atur `binlog_row_image` variabel ke. FULL Pengaturan ini memastikan bahwa DMS menerima informasi yang cukup untuk membangun Data Manipulation Language (DHTML) lengkap untuk database target selama migrasi.

Untuk mengatur gambar binlogFULL, lakukan hal berikut:

- Untuk Amazon RDS, nilai ini secara FULL default.
- Untuk database yang menggunakan selang lokal atau di Amazon EC2, tetapkan nilainya `binlog_row_image` di (`my.ini`Microsoft Windows) atau (UNIX). `my.cnf`

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Untuk informasi selengkapnya tentang server MariaDB yang dihosting sendiri, lihat. [Menggunakan database yang kompatibel dengan MySQL yang dikelola sendiri sebagai sumber AWS DMS](#)

Validasi jika database sumber adalah MariaDB Read-Replica

Kunci API: `mariadb-check-database-role`

Penilaian premi ini memverifikasi apakah database sumber adalah replika baca. Untuk mengaktifkan dukungan CDC untuk DMS saat terhubung ke replika baca, atur parameter ke. `log_slave_updates True` Untuk informasi selengkapnya tentang menggunakan database MySQL yang dikelola sendiri, lihat. [Menggunakan database yang kompatibel dengan MySQL yang dikelola sendiri sebagai sumber AWS DMS](#)

Untuk mengatur `log_slave_updates` nilainyaTrue, lakukan hal berikut:

- Untuk Amazon RDS, gunakan grup parameter database. Untuk informasi tentang menggunakan grup parameter database RDS, lihat [Bekerja dengan grup parameter](#) di Panduan Pengguna Amazon RDS.
- Untuk database yang menggunakan selang lokal atau di Amazon EC2, tetapkan nilainya `log_slave_updates` di (`my.ini`Microsoft Windows) atau (UNIX). `my.cnf`

Penilaian ini hanya berlaku untuk migrasi muatan penuh dan CDC, atau migrasi khusus CDC. Penilaian ini tidak berlaku untuk migrasi muatan penuh saja.

Validasi jika tabel memiliki partisi, dan merekomendasikan **TRUNCATE_BEFORE_LOAD** atau **DO_NOTHING** untuk pengaturan tugas beban penuh

Kunci API: mariadb-check-table-partition

Penilaian premi ini memeriksa keberadaan tabel dengan partisi dalam database sumber. DMS membuat tabel tanpa partisi pada target MariaDB. Untuk memigrasikan tabel yang dipartisi ke tabel yang dipartisi pada target, Anda harus melakukan hal berikut:

- Pra-buat tabel yang dipartisi dalam database MariaDB target.
- Konfigurasi tugas DMS Anda untuk digunakan TRUNCATE_BEFORE_LOAD atau DO_NOTHING sebagai pengaturan tugas beban penuh.

Untuk informasi selengkapnya tentang batasan titik akhir MariaDB, lihat. [Batasan dalam menggunakan database MySQL sebagai sumber AWS DMS](#)

Validasi jika DMS mendukung versi database

Kunci API: mariadb-check-supported-version

Penilaian premi ini memverifikasi apakah versi database sumber kompatibel dengan DMS. CDC tidak didukung dengan Amazon RDS MariaDB versi 10.4 atau lebih rendah, atau versi MySQL yang lebih besar dari 10.11. Untuk informasi selengkapnya tentang versi MariaDB yang didukung, lihat. [Titik akhir sumber untuk migrasi data](#)

Validasi jika database target dikonfigurasi untuk disetel **local_infile** ke 1

Kunci API: mariadb-check-target-localinfile-set

Penilaian premi ini memeriksa apakah `local_infile` parameter dalam database target diatur ke 1. DMS memerlukan parameter 'local_infile' untuk disetel ke 1 selama pemuatan penuh dalam database target Anda. Untuk informasi selengkapnya, lihat [Migrasi dari MySQL ke MySQL menggunakan AWS DMS](#).

Penilaian ini hanya berlaku untuk tugas beban penuh.

Validasi jika database target memiliki tabel dengan kunci asing

Kunci API: mariadb-check-fk-target

Penilaian premi ini memeriksa apakah tugas penuh atau penuh dan tugas CDC yang bermigrasi ke database MariaDB memiliki tabel dengan kunci asing. Pengaturan default dalam DMS adalah memuat tabel dalam urutan abjad. Tabel dengan kunci asing dan batasan integritas referensial dapat menyebabkan beban gagal, karena tabel induk dan anak mungkin tidak dimuat pada saat yang bersamaan.

Untuk informasi selengkapnya tentang integritas referensial di DMS, lihat [Bekerja dengan indeks, pemicu, dan kendala integritas referensial dalam topik. Meningkatkan performa suatu migrasi AWS DMS](#)

Validasi jika tabel sumber dalam lingkup tugas memiliki kendala kaskade

Kunci API: `mariadb-check-cascade-constraints`

Penilaian premi ini memeriksa apakah ada tabel sumber MariaDB yang memiliki kendala kaskade. Batasan kaskade tidak dimigrasikan atau direplikasi oleh tugas DMS, karena MariaDB tidak mencatat perubahan untuk peristiwa ini di binlog. Meskipun AWS DMS tidak mendukung kendala ini, Anda dapat menggunakan solusi untuk target database relasional.

Untuk informasi tentang mendukung batasan cascade dan kendala lainnya, lihat [Indeks, Kunci Asing, atau Pembaruan atau Penghapusan Cascade Tidak Dimigrasi](#) di topik tugas migrasi pemecahan masalah. AWS DMS

Validasi jika tabel sumber dalam lingkup tugas telah menghasilkan kolom

Kunci API: `mariadb-check-generated-columns`

Penilaian premi ini memeriksa apakah salah satu tabel sumber MariaDB telah menghasilkan kolom. Tugas DMS tidak memigrasi atau mereplikasi kolom yang dihasilkan.

Untuk informasi tentang cara memigrasi kolom yang dihasilkan, lihat [????](#).

Validasi jika nilai batas waktu sesuai untuk sumber MariaDB

Kunci API: `mariadb-check-source-network-parameter`

Penilaian premi ini memeriksa apakah titik akhir sumber MariaDB tugas memilikinet_read_timeout, net_wait_timeout dan wait_timeout pengaturan disetel ke setidaknya 300 detik. Ini diperlukan untuk mencegah pemutusan selama migrasi.

Untuk informasi selengkapnya, lihat [Koneksi ke instans MySQL target terputus selama pengerjaan tugas.](#)

Validasi jika nilai batas waktu sesuai untuk target MariaDB

Kunci API: mariadb-check-target-network-parameter

Penilaian premi ini memeriksa apakah titik akhir target MariaDB tugas memilikinet_read_timeout, net_wait_timeout dan wait_timeout pengaturan disetel ke setidaknya 300 detik. Ini diperlukan untuk mencegah pemutusan selama migrasi.

Untuk informasi selengkapnya, lihat [Koneksi ke instans MySQL target terputus selama pengerjaan tugas](#).

Penilaian PostgreSQL

Bagian ini menjelaskan penilaian premi individu untuk tugas migrasi yang menggunakan titik akhir sumber PostgreSQL.

Topik

- [Validasi jika versi basis data sumber didukung oleh DMS untuk migrasi](#)
- [Validasi logical_decoding_work_mem parameter pada database sumber](#)
- [Validasi apakah database sumber memiliki transaksi yang berjalan lama](#)
- [Validasi parameter basis data sumber max_slot_wal_keep_size](#)
- [Periksa apakah parameter database sumber postgres-check-maxwalsenders diatur untuk mendukung CDC.](#)
- [Periksa apakah database sumber dikonfigurasi untuk PGLOGICAL](#)
- [Validasi jika kunci primer tabel sumber adalah dari LOB Datatype](#)
- [Validasi jika tabel sumber memiliki kunci primer](#)
- [Validasi jika transaksi yang disiapkan ada di database sumber](#)
- [Validasi jika wal_sender_timeout diatur ke nilai minimum yang diperlukan untuk mendukung DMS CDC](#)
- [Validasi jika wal_level diatur ke logis pada database sumber](#)

Validasi jika versi basis data sumber didukung oleh DMS untuk migrasi

Kunci API: postgres-check-dbversion

Penilaian premi ini memverifikasi apakah versi basis data sumber kompatibel dengan. AWS DMS

Validasi **logical_decoding_work_mem** parameter pada database sumber

Kunci API: `postgres-check-for-logical-decoding-work-mem`

Penilaian premi ini merekomendasikan penyetelan `logical_decoding_work_mem` parameter pada database sumber. Pada database yang sangat transaksional di mana Anda mungkin memiliki transaksi yang berjalan lama atau banyak sub-transaksi, ini dapat mengakibatkan peningkatan konsumsi memori decoding logis dan kebutuhan untuk tumpah ke disk. Ini menghasilkan latensi sumber DMS yang tinggi selama replikasi. Dalam skenario seperti itu, Anda mungkin perlu menyetel `logical_decoding_work_mem`. Parameter ini didukung di PostgreSQL versi 13 dan lebih besar.

Validasi apakah database sumber memiliki transaksi yang berjalan lama

Kunci API: `postgres-check-longrunningtxn`

Penilaian premi ini memverifikasi apakah database sumber memiliki transaksi berjalan lama yang berlangsung lebih dari 10 menit. Memulai tugas mungkin gagal, karena secara default, DMS memeriksa setiap transaksi terbuka saat memulai tugas.

Validasi parameter basis data sumber **max_slot_wal_keep_size**

Kunci API: `postgres-check-maxslot-wal-keep-size`

Penilaian premi ini memverifikasi nilai yang dikonfigurasi untuk `max_slot_wal_keep_size`. Ketika `max_slot_wal_keep_size` diatur ke nilai non-default, tugas DMS mungkin gagal karena penghapusan file WAL yang diperlukan.

Periksa apakah parameter database sumber **postgres-check-maxwalsenders** diatur untuk mendukung CDC.

Kunci API: `postgres-check-maxwalsenders`

Penilaian premi ini memverifikasi nilai yang dikonfigurasi untuk database `max_wal_senders` sumber. DMS harus diatur lebih besar dari 1 untuk mendukung Change Data Capture (CDC). `max_wal_senders`

Periksa apakah database sumber dikonfigurasi untuk **PGLLOGICAL**

Kunci API: `postgres-check-pglogical`

Penilaian premi ini memverifikasi apakah `shared_preload_libraries` nilai disetel ke dukungan `pglogical` PLOGICAL untuk CDC. Perhatikan bahwa Anda dapat mengabaikan penilaian ini jika Anda berencana menggunakan decoding pengujian untuk replikasi logis.

Validasi jika kunci primer tabel sumber adalah dari LOB Datatype

Kunci API: `postgres-check-pk-lob`

Penilaian premi ini memverifikasi apakah kunci utama tabel adalah tipe data Objek Besar (LOB). DMS tidak mendukung replikasi jika tabel sumber memiliki kolom LOB sebagai kunci utama.

Validasi jika tabel sumber memiliki kunci primer

Kunci API: `postgres-check-pk`

Penilaian premi ini memverifikasi apakah kunci utama ada untuk tabel yang digunakan dalam lingkup tugas. DMS tidak mendukung replikasi untuk tabel tanpa kunci utama, kecuali identitas replika diatur ke `full` pada tabel sumber.

Validasi jika transaksi yang disiapkan ada di database sumber

Kunci API: `postgres-check-preparedtxn`

Penilaian premi ini memverifikasi apakah ada transaksi yang disiapkan yang ada di database sumber. Pembuatan slot replikasi mungkin berhenti merespons jika ada transaksi yang disiapkan pada basis data sumber.

Validasi jika `wal_sender_timeout` diatur ke nilai minimum yang diperlukan untuk mendukung DMS CDC

Kunci API: `postgres-check-walsendersttimeout`

Penilaian premi ini memverifikasi jika `wal_sender_timeout` disetel ke minimum 10000 milidetik (10 detik). Tugas DMS dengan CDC membutuhkan minimal 10.000 milidetik (10 detik), dan gagal jika nilainya kurang dari 10.000.

Validasi jika `wal_level` diatur ke logis pada database sumber

Kunci API: `postgres-check-wallevel`

Penilaian premi ini memverifikasi jika disetel ke `wal_level` logis. Agar DMS CDC berfungsi, parameter ini perlu diaktifkan pada database sumber.

Memulai dan melihat penilaian tipe data (Legacy)

Note

Bagian ini menjelaskan konten warisan. Kami menyarankan Anda menggunakan proses penilaian premi, yang dijelaskan sebelumnya di [Menentukan, memulai, dan melihat penilaian perdana berjalan](#)

Penilaian tipe data tidak tersedia di konsol. Anda hanya dapat menjalankan penilaian tipe data menggunakan API atau CLI, dan Anda hanya dapat melihat hasil penilaian tipe data di bucket S3 tugas.

Penilaian tipe data mengidentifikasi tipe data dalam database sumber yang mungkin tidak dimigrasi dengan benar karena target tidak mendukungnya. Selama penilaian ini, AWS DMS membaca skema database sumber untuk tugas migrasi dan membuat daftar tipe data kolom. Kemudian membandingkan daftar ini dengan daftar tipe data yang telah ditentukan yang didukung oleh AWS DMS. Jika tugas migrasi Anda memiliki tipe data yang tidak didukung, AWS DMS buat laporan yang dapat Anda lihat untuk melihat apakah tugas migrasi Anda memiliki tipe data yang tidak didukung. AWS DMS tidak membuat laporan jika tugas migrasi Anda tidak memiliki tipe data yang tidak didukung.

AWS DMS mendukung pembuatan laporan penilaian tipe data untuk database relasional berikut:

- Oracle
- SQL Server
- PostgreSQL
- MySQL
- MariaDB
- Amazon Aurora

Anda dapat memulai dan melihat laporan penilaian tipe data menggunakan CLI dan SDK untuk mengakses API: AWS DMS

- CLI menggunakan [start-replication-task-assessment](#) perintah untuk memulai penilaian tipe data dan menggunakan [describe-replication-task-assessment-results](#) perintah untuk melihat laporan penilaian tipe data terbaru dalam format JSON.

- AWS DMS API menggunakan [StartReplicationTaskAssessment](#) operasi untuk memulai penilaian tipe data dan menggunakan [DescribeReplicationTaskAssessmentResults](#) operasi untuk melihat laporan penilaian tipe data terbaru dalam format JSON.

Laporan penilaian tipe data adalah file JSON tunggal yang mencakup ringkasan yang berisi daftar jenis data yang tidak didukung dan jumlah kolom untuk masing-masing. Ini termasuk daftar struktur data untuk setiap jenis data yang tidak didukung termasuk skema, tabel, dan kolom yang memiliki tipe data yang tidak didukung. Anda dapat menggunakan laporan untuk memodifikasi jenis data sumber dan meningkatkan keberhasilan migrasi.

Ada dua tingkat tipe data yang tidak didukung. Jenis data yang muncul di laporan yang tidak didukung tidak dapat dimigrasi. Jenis data yang muncul di laporan sebagai sebagian didukung mungkin dikonversi ke tipe data lain, tetapi tidak bermigrasi seperti yang Anda harapkan.

Contoh berikut menunjukkan contoh laporan penilaian tipe data yang mungkin Anda lihat.

```
{
  "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 menyimpan penilaian tipe data terbaru dan sebelumnya di bucket Amazon S3 yang dibuat AWS DMS oleh di akun Anda. Nama bucket Amazon S3 memiliki format berikut, di mana *customerId* adalah ID pelanggan Anda dan *CustomerDNS* adalah pengidentifikasi internal.

```
dms-customerId-customerDNS
```

Note

Secara default, Anda dapat membuat hingga 100 bucket Amazon S3 di setiap akun AWS Anda. Karena AWS DMS membuat bucket di akun Anda, pastikan itu tidak melebihi batas bucket Anda. Jika tidak, penilaian tipe data gagal.

Semua laporan penilaian tipe data untuk tugas migrasi tertentu disimpan dalam folder bucket yang diberi nama dengan pengidentifikasi tugas. Setiap nama file laporan adalah tanggal penilaian tipe data dalam format yyyy-mm-dd-hh -mm. Anda dapat melihat dan membandingkan laporan penilaian tipe data sebelumnya dari Konsol Manajemen Amazon S3.

AWS DMS juga membuat peran AWS Identity and Access Management (IAM) untuk memungkinkan akses ke bucket S3 yang dibuat untuk laporan ini. Nama perannya adalah `dms-access-for-tasks`. Peran menggunakan `AmazonDMSRedshiftS3RoleKebijakan`. Jika terjadi `ResourceNotFoundFault` kesalahan saat Anda menjalankan `StartReplicationTaskAssessment`, lihat [ResourceNotFoundFault](#) di bagian Pemecahan Masalah untuk informasi tentang membuat `dms-access-for-tasks` peran secara manual.

Penilaian pemecahan masalah berjalan

Setelah itu, Anda dapat menemukan topik tentang pemecahan masalah dengan menjalankan laporan penilaian dengan. AWS Database Migration Service Topik-topik ini dapat membantu Anda menyelesaikan masalah umum.

Topik

- [ResourceNotFoundFault saat berlari StartReplicationTaskAssessment](#)

ResourceNotFoundFault saat berlari StartReplicationTaskAssessment

Anda mungkin menemukan pengecualian berikut saat menjalankan [StartReplicationTaskAssessment](#) tindakan.

```
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
```

Jika Anda menemukan pengecualian ini, buat `dms-access-for-tasks` peran dengan melakukan hal berikut:

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Peran.
3. Pilih Buat peran.
4. Pada halaman Pilih entitas tepercaya, untuk jenis entitas tepercaya, pilih Kebijakan kepercayaan khusus.
5. Tempel JSON berikut di editor, ganti teks yang ada.

```
{
  "Version": "2012-10-17",
```

```
"Statement": [  
  {  
    "Sid": "1",  
    "Effect": "Allow",  
    "Principal": {  
      "Service": "dms.amazonaws.com"  
    },  
    "Action": "sts:AssumeRole"  
  }  
]
```

Kebijakan sebelumnya memberikan izin untuk `sts:AssumeRole` AWS DMS Saat Anda menambahkan kebijakan `AmazonDMSredShifts3role`, DMS dapat membuat bucket S3 di akun Anda, dan memasukkan hasil penilaian tipe data ke dalam bucket S3 ini.

6. Pilih Berikutnya.
7. Pada halaman Tambahkan izin, cari dan tambahkan kebijakan `AmazonDMSredShifts3role`. Pilih Berikutnya.
8. Pada halaman Nama, tinjau, dan buat, beri nama peran `dms-access-for-tasks`. Pilih Buat peran.

Menentukan data tambahan untuk pengaturan tugas

Ketika Anda membuat atau memodifikasi tugas replikasi untuk beberapa AWS DMS, tugas mungkin memerlukan informasi tambahan untuk melakukan migrasi. Anda dapat menentukan informasi tambahan ini menggunakan opsi di konsol DMS. Atau Anda dapat menentukan menggunakan parameter `TaskData` untuk operasi DMS API `CreateReplicationTask` atau `ModifyReplicationTask`.

Jika titik akhir target Anda adalah Amazon Neptune, Anda perlu menentukan data pemetaan, tambahan untuk pemetaan tabel. Data pemetaan tambahan ini menentukan bagaimana mengkonversi sumber data relasional ke data grafik target yang database Neptune dapat mengkonsumsi. Dalam kasus ini, Anda dapat menggunakan salah satu dari dua format yang mungkin. Untuk informasi selengkapnya, lihat [Menentukan aturan pemetaan grafik menggunakan Gremlin dan R2RLL untuk Amazon Neptune sebagai target](#).

Pemantauan Tugas AWS DMS

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan performa AWS DMS dan solusi AWS Anda. Anda harus mengumpulkan data pemantauan dari semua bagian solusi AWS Anda sehingga Anda dapat lebih mudah mengidentifikasi dan menghapus kesalahan pada banyak poin yang gagal jika hal itu terjadi. AWS Menyediakan beberapa alat untuk pemantauan tugas dan sumber daya AWS DMS Anda dan merespons terhadap potensi insiden:

Tindakan dan notifikasi AWS DMS

AWS DMS menggunakan Amazon Simple Notification Service (Amazon SNS) untuk memberikan notifikasi pada saat tindakan AWS DMS terjadi, misalnya pembuatan atau penghapusan instans replikasi. AWS DMS mengelompokkan sejumlah tindakan ke dalam beberapa kategori untuk Anda dapat berlangganan, sehingga Anda dapat menerima pemberitahuan di saat suatu tindakan dalam kategori tersebut terjadi. Misalnya, jika Anda berlangganan kategori Pembuatan untuk suatu instans replikasi tertentu, Anda akan diberi tahu setiap kali ada tindakan terkait pembuatan yang memengaruhi instans replikasi Anda. Anda dapat bekerja dengan notifikasi ini dalam bentuk apa pun yang didukung oleh Amazon SNS untuk Wilayah AWS, seperti pesan email, pesan teks, atau panggilan ke titik akhir HTTP. Untuk informasi selengkapnya, lihat [Bekerja dengan acara Amazon SNS dan pemberitahuan di AWS Database Migration Service](#)

Status Tugas

Anda dapat memantau kemajuan tugas Anda dengan memeriksa status tugas dan memantau tabel kontrol tugas tersebut. Status tugas menunjukkan syarat tugas AWS DMS dan sumber daya yang terkait. Ini termasuk indikasi seperti jika tugas sedang dibuat, mulai, berjalan, atau berhenti. Status ini juga mencakup keadaan saat tabel dengan tugas dilakukan migrasi, seperti jika beban penuh dari suatu tabel telah dimulai atau sedang berlangsung dan detail seperti jumlah sisipan, penghapusan, dan pembaruan telah berlangsung untuk tabel tersebut. Untuk informasi selengkapnya tentang pemantauan syarat tugas dan sumber daya tugas, lihat [Status Tugas](#) dan [Kondisi tabel selama tugas](#). Untuk informasi selengkapnya tentang tabel pengendali, lihat [Mengontrol pengaturan tugas tabel](#).

Amazon CloudWatch alarm dan log log alarm

Menggunakan Amazon CloudWatch alarm, Anda melihat satu metrik tugas selama periode waktu yang Anda tentukan. Jika metrik melebihi ambang batas tertentu, notifikasi dikirimkan ke topik Amazon SNS. CloudWatch Alarm tidak termasuk beberapa statistik karena mereka berada dalam beberapa statistik tertentu. Sebaliknya, kondisi harus diubah dan dipertahankan untuk beberapa

periode tertentu. AWS DMS juga menggunakan CloudWatch untuk beberapa statistik untuk beberapa statistik untuk beberapa statistik. Anda dapat menggunakan AWS CLI atau AWS DMS API untuk melihat informasi tentang catatan tugas. Untuk beberapa statistik tentang penggunaan CloudWatch bersama AWS DMS, lihat [Pemantauan tugas replikasi menggunakan Amazon CloudWatch](#). Untuk informasi selengkapnya tentang cara pemantauan metrik AWS DMS, lihat [Metrik AWS Database Migration Service](#). Untuk informasi lebih lanjut tentang penggunaan catatan tugas AWS DMS, lihat [Melihat dan mengelola AWS Mencatat tugas DMS](#).

Log Perjalanan Waktu

Untuk mencatat dan men-debug tugas replikasi, Anda dapat menggunakan AWS DMS Perjalanan Waktu. Dalam pendekatan ini, Anda menggunakan Amazon S3 untuk menyimpan log dan mengenkripsi mereka menggunakan kunci enkripsi Anda. Anda dapat mengambil log S3 Anda menggunakan filter tanggal-waktu, lalu melihat, mengunduh, dan mengaburkan log sesuai kebutuhan. Dengan melakukan ini, Anda dapat “melakukan perjalanan kembali ke masa lalu” untuk menyelidiki aktivitas database.

Anda dapat menggunakan Time Travel dengan titik akhir sumber PostgreSQL yang didukung DMS dan PostgreSQL yang didukung DMS dan titik akhir target PostgreSQL dan MySQL. Anda dapat mengaktifkan Perjalanan Waktu hanya untuk tugas beban penuh dan CDC dan untuk tugas khusus CDC. Untuk mengaktifkan Perjalanan Waktu atau mengubah pengaturan Perjalanan Waktu yang ada, pastikan tugas Anda dihentikan.

Untuk beberapa statistik tentang beberapa statistik perjalanan waktu, lihat [Pengaturan tugas Perjalanan Waktu](#). Untuk praktik terbaik dalam menggunakan log Perjalanan Waktu, lihat [Memecahkan masalah tugas replikasi dengan Time Travel](#).

Log AWS CloudTrail

AWS DMS terintegrasi dengan AWS CloudTrail layanan yang memberikan catatan tindakan yang dilakukan oleh pengguna, peran IAM, AWS layanan di AWS DMS. CloudTrail menangkap semua panggilan API untuk AWS DMS sebagai beberapa peristiwa, termasuk beberapa statistik AWS DMS konsol dan dari panggilan kode ke AWS DMS Operasi API. Jika Anda membuat jejak, Anda dapat mengaktifkan beberapa statistik CloudTrail peristiwa ke beberapa bucket Amazon S3, termasuk beberapa acara untuk beberapa statistik AWS DMS. Jika Anda ingin membuat jejak, Anda dapat melihat peristiwa terbaru CloudTrail konsol di Sejarah acara. Menggunakan informasi yang dikumpulkan oleh CloudTrail Anda dapat menentukan beberapa permintaan untuk beberapa statistik AWS DMS, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan,

kapan permintaan, dan detail tambahan. Untuk informasi selengkapnya, lihat [Mencatat panggilan API AWS DMS dengan AWS CloudTrail](#).

Mencatat Basis Data

Anda dapat melihat, mengunduh, dan melihat catatan basis data untuk titik akhir tugas menggunakan AWS Management Console, AWS CLI, atau API untuk layanan basis data AWS. Untuk informasi selengkapnya, lihat dokumentasi untuk layanan basis data Anda di [dokumentasi AWS](#).

Untuk informasi lebih lanjut, lihat topik berikut.

Topik

- [Status Tugas](#)
- [Kondisi tabel selama tugas](#)
- [Pemantauan tugas replikasi menggunakan Amazon CloudWatch](#)
- [Metrik AWS Database Migration Service](#)
- [Melihat dan mengelola AWSMencatat tugas DMS](#)
- [Mencatat panggilan API AWS DMS dengan AWS CloudTrail](#)
- [AWS DMSEndit log koneksi](#)

Status Tugas

Status tugas menunjukkan kondisi tugas. Tabel berikut menunjukkan status yang mungkin dapat dimiliki suatu tugas:

Status Tugas	Deskripsi
Menciptakan	AWS DMS sedang membuat tugas.
Berlari	Tugas tersebut adalah melaksanakan tugas migrasi yang ditentukan.
Berhenti	Tugas dihentikan.
Berhenti	Tugas sedang dihentikan. Ini biasanya merupakan indikasi intervensi pengguna dalam tugas.

Status Tugas	Deskripsi
Menghapus	Tugas sedang dihapus, biasanya dari permintaan untuk campur tangan pengguna.
Gagal	Tugas telah gagal. Untuk informasi lebih lanjut, lihat file yang mencatat tugas.
Kesalahan	Tugas berhenti karena beberapa kesalahan. Deskripsi singkat tentang kesalahan tugas disediakan di bagian pesan kegagalan terakhir. Lihat kesalahan .
Berjalan dengan kesalahan	Tugas berjalan dengan status kesalahan. Ini biasanya menunjukkan bahwa satu tabel dalam tugas tidak dapat dimigrasi. Tugas terus memuat tabel lain sesuai dengan aturan pemilihan.
Memulai	Tugas ini menjadikan connect ke instans replikasi dan ke titik akhir sumber dan target. Setiap filter dan transformasi sedang diterapkan.
Siap	Tugas ini siap dijalankan. Status ini biasanya mengikuti status “membuat”.
Memodifikasi	Tugas sedang diubah, biasanya disebabkan oleh tindakan pengguna yang mengubah pengaturan tugas.
Bergerak	Tugas ini sedang dalam proses dipindahkan ke instans replikasi lain. Replikasi tetap berada dalam keadaan ini sampai langkah ini selesai. Menghapus tugas adalah satu-satunya operasi diizinkan pada tugas replikasi sementara tugas itu sedang dipindahkan.
Gerakan gagal	Pemindahan tugas telah gagal untuk alasan apa pun, seperti halnya tidak memiliki cukup ruang penyimpanan pada instans replikasi target. Ketika tugas replikasi berada dalam keadaan ini, maka tugas ini dapat dimulai, dimodifikasi, dipindahkan, atau dihapus.

Status Tugas	Deskripsi
Pengujian	Migrasi database yang ditentukan untuk tugas ini sedang diuji sebagai respons terhadap menjalankan salah satu StartReplicationTaskAssessmentRun , atau StartReplicationTaskAssessment operasi

Bilah status tugas memberikan estimasi tentang kemajuan tugas. Kualitas estimasi ini tergantung pada kualitas statistik tabel basis data sumber; semakin baik statistik tabel, semakin akurat estimasi. Untuk tugas dengan hanya satu tabel yang tidak memiliki estimasi statistik baris, kita tidak dapat memberikan persentase estimasi lengkap apa pun. Dalam hal ini, gunakan status tugas dan indikasi baris yang dimuat untuk mengonfirmasi bahwa tugas berlangsung dan mengalami kemajuan.

Perhatikan bahwa kolom “terakhir diperbarui” di konsol DMS hanya menunjukkan waktu AWS DMS terakhir yang memperbarui catatan statistik tabel untuk suatu tabel. Ini tidak menunjukkan waktu pembaruan terakhir pada tabel.

Selain menggunakan konsol DMS, Anda bisakeluarandeskripsi tugas replikasi saat ini, termasuk status tugas, dengan menggunakan `aws dms describe-replication-tasks` perintah di [AWS CLI](#), seperti yang ditunjukkan pada contoh berikut.

```
{
  "ReplicationTasks": [
    {
      "ReplicationTaskIdentifier": "moveit2",
      "SourceEndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWYGAYUVLKIB732KEVWA",
      "TargetEndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",
      "ReplicationInstanceArn": "arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
      "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
    }
  }
]
}

```

Kondisi tabel selama tugas

Konsol DMS AWS memperbarui informasi mengenai keadaan tabel Anda selama migrasi. Tabel berikut menunjukkan kemungkinan nilai keadaan:

The screenshot shows the AWS DMS console interface for a migration task named 'dms-gs-task'. The 'Table statistics' tab is selected and highlighted with a red box. Below the tab, there is a search bar for 'Find schema' and a table with 157 rows. The table has columns for 'Schema name', 'Table', 'Load state', and 'Elapsed load time'. The 'Load state' column is highlighted with a red box, showing values such as 'Table error' and 'Table completed'. The 'Elapsed load time' column shows values like '< 1 s'.

Schema name	Table	Load state	Elapsed load time
mysql	user	Table error	< 1 s
mysql	server_cost	Table completed	< 1 s
mysql	tables_priv	Table completed	< 1 s
mysql	gtid_executed	Table completed	< 1 s
mysql	replication_asynchronous_connection_failover	Table completed	< 1 s

Kondisi	Deskripsi
Tabel tidak ada	AWS DMS tidak dapat menemukan tabel pada titik akhir sumber.
Sebelum memuat	Mengaktifkan proses beban penuh telah dilakukan, tetapi belum mulai.
Beban penuh	Proses beban penuh sedang berlangsung.
Tabel selesai	Beban penuh telah selesai.
Tabel dibatalkan	Pemuatan beban tabel telah dibatalkan.
Kesalahan tabel	Terjadi kesalahan saat memuat beban tabel.

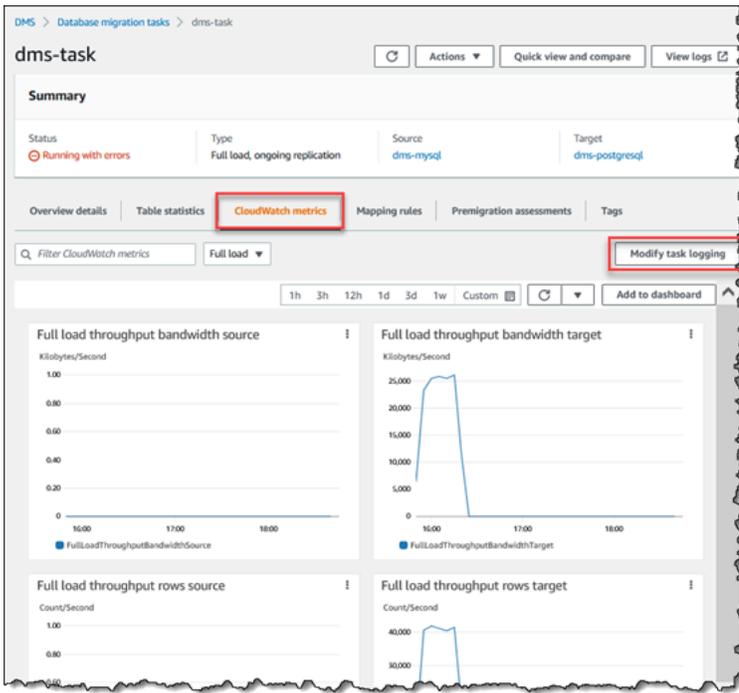
Pemantauan tugas replikasi menggunakan Amazon CloudWatch

Anda dapat menggunakan Amazon CloudWatch alarm atau beberapa statistik untuk beberapa statistik untuk beberapa statistik untuk beberapa statistik Untuk beberapa statistik tentang Amazon CloudWatch, lihat [Apa itu Amazon CloudWatch, Amazon CloudWatch Acara, dan Amazon CloudWatch Log?](#) di Amazon CloudWatch Panduan pengguna. Perhatikan bahwa ada biaya untuk beberapa statistik untuk beberapa statistik CloudWatch.

Jika tugas replikasi Anda tidak dibuat CloudWatch log, lihat [AWS DMS tidak membuat CloudWatch log](#) dalam panduan pemecahan masalah.

The AWS DMS konsol menunjukkan dasar CloudWatch statistik untuk setiap tugas, termasuk status tugas, persen selesai, waktu yang telah berlalu, dan statistik tabel, seperti yang ditunjukkan berikut. Pilih tugas replikasi dan kemudian pilih CloudWatch metrik tab.

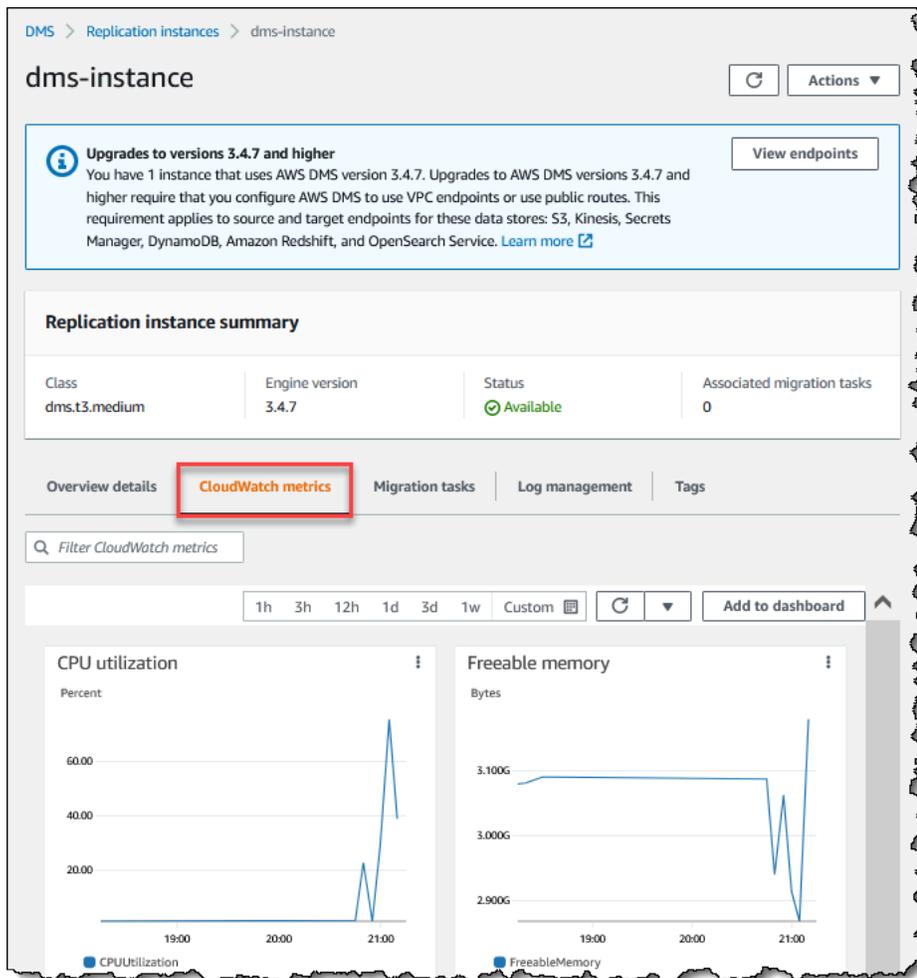
Untuk melihat dan memodifikasi CloudWatch pengaturan log tugas, pilih Udit log tugas. Untuk informasi selengkapnya, lihat [Pengaturan tugas pengelogan](#).



Konsol AWS DMS menunjukkan statistik performa untuk setiap tabel, termasuk jumlah sisipan, penghapusan, dan pembaruan, ketika Anda menentukan pilihan tab Statistik tabel.

Schema name	Table	Load state	Elapsed load time	Inserts	Deletes	Updates	DDLs	Applied inserts
mysql	user	Table error	1 s	0	0	0	0	0
mysql	server_cost	Table completed	1 s	0	0	0	0	0
dms_sample	seat_type	Table completed	< 1 s	0	0	0	0	0
mysql	tables_priv	Table completed	1 s	0	0	0	0	0

Selain itu, jika Anda memilih instance replikasi dari halaman, Anda dapat melihat metrik kinerja untuk contoh dengan memilih CloudWatch metrik tab.



Metrik AWS Database Migration Service

AWS DMS menyediakan statistik untuk hal berikut:

- Metrik Host— Statistik kinerja dan pemanfaatan untuk host, yang disediakan oleh Amazon CloudWatch. Untuk daftar lengkap metrik yang tersedia, lihat [Metrik instans replikasi](#).
- Metrik Tugas Replikasi – Statistik untuk tugas replikasi termasuk perubahan masuk dan yang dilakukan, dan latensi antara host replikasi dan basis data sumber dan target. Untuk daftar lengkap metrik yang tersedia, lihat [Metrik tugas replikasi](#).
- Metrik tabel – Statistik untuk tabel yang sedang dalam proses migrasi, termasuk jumlah sisipan, pembaruan, penghapusan, dan pernyataan DDL yang selesai.

Metrik tugas dibagi menjadi beberapa statistik antara host replikasi dan titik akhir sumber, dan statistik antara host replikasi dan titik akhir target. Anda dapat menentukan total statistik untuk suatu

tugas dengan menambahkan dua statistik terkait bersama-sama. Misalnya, Anda dapat menentukan latensi total, atau kelambatan, untuk tugas `CDCLatencySource` dan `CDCLatencyTarget` nilai.

Nilai metrik tugas dapat dipengaruhi oleh aktivitas saat ini di dalam basis data sumber Anda. Misalnya, jika transaksi telah dimulai, tetapi belum dilakukan, `CDCLatencySource` metrik terus tumbuh hingga transaksi tersebut telah dilakukan.

Untuk contoh replikasi, `FreeableMemory` metrik membutuhkan klarifikasi. Memori bebas bukan indikasi dari memori bebas sebenarnya yang tersedia. Ini merupakan memori yang saat ini digunakan yang dapat dibebaskan dan digunakan untuk kegunaan lain; ini merupakan kombinasi dari buffer dan cache yang digunakan pada instans replikasi.

Sementara `FreeableMemory` metrik tidak mencerminkan memori bebas aktual yang tersedia, kombinasi dari `FreeableMemory` dan `SwapUsage` metrik dapat menunjukkan apakah instance replikasi kelebihan beban.

Memantau kedua metrik ini untuk syarat-syarat berikut:

- The `FreeableMemory` metrik mendekati nol
- The `SwapUsage` metrik meningkat atau berfluktuasi

Jika Anda melihat salah satu dari dua syarat ini, itu menunjukkan bahwa Anda harus mempertimbangkan pindah ke instans replikasi yang lebih besar. Anda juga harus mempertimbangkan untuk mengurangi jumlah dan tipe tugas yang berlangsung pada instans replikasi. Tugas beban penuh memerlukan lebih banyak memori daripada tugas yang hanya melakukan replikasi perubahan.

Untuk memperkirakan secara kasar kebutuhan memori aktual untuk AWS DMS tugas migrasi, Anda dapat menggunakan beberapa parameter berikut.

Kolom LOB

Jumlah rata-rata kolom LOB di setiap tabel dalam lingkup migrasi Anda.

Jumlah maksimum tabel untuk beberapa statistik untuk beberapa statistik untuk beberapa statistik

Jumlah maksimum tabel yang AWS DMS memuat secara paralel dalam satu tugas.

Nilai default adalah 8.

Ukuran potongan LOB

Ukuran potongan LOB, dalam kilobyte, ituAWS DMSdigunakan untuk mereplikasi data ke database target.

Tingkat komit selama beberapa beban

Jumlah maksimum beberapa statistik yangAWS DMSdapat mentransfer secara paralel.

Nilai default adalah 10,000.

Ukuran LOB

Ukuran maksimum LOB individu, dalam beberapa kilobyte.

Ukuran array massal

Jumlah maksimum baris yang diambil atau diproses oleh driver endpoint Anda. Nilai ini tergantung pada beberapa pengaturan driver.

Nilai default adalah 1,000.

Setelah Anda menentukan nilai ini, Anda dapat menggunakan salah satu metode berikut untuk memperkirakan jumlah memori Anda. Metode ini tergantung pada opsi yang Anda pilihPengaturan kolom LOBdalam tugas migrasi Anda.

- UntukMode LOB penuh, gunakan rumus berikut.

$$\text{Required memory} = (\text{LOB columns}) * (\text{Maximum number of tables to load in parallel}) * (\text{LOB chunk size}) * (\text{Commit rate during full load})$$

Pertimbangkan contoh di mana tabel sumber Anda menyertakan rata-rata 2 kolom LOB, dan ukuran potongan LOB adalah 64 KB. Jika Anda menggunakan beberapa nilai untuk beberapa statistikMaximum number of tables to load in paralleldanCommit rate during full load, maka jumlah memori yang dibutuhkan untuk tugas Anda adalah sebagai berikut.

$$\text{Required memory} = 2 * 8 * 64 * 10,000 = 10,240,000 \text{ KB}$$

Note

Untuk mengurangi nilaiTingkat komit selama beberapa beban, bukaAWS DMSkonsol, pilihTugas migrasi database, dan membuat atau memodifikasi tugas. PerluasPengaturan lanjutan, dan masukkan nilai Anda untukTingkat komit selama beberapa beban.

- UntukMode LOB terbatas, gunakan rumus berikut.

$$\text{Required memory} = (\text{LOB columns}) * (\text{Maximum number of tables to load in parallel}) * (\text{LOB size}) * (\text{Bulk array size})$$

Pertimbangkan contoh di mana tabel sumber Anda menyertakan rata-rata 2 kolom LOB, dan ukuran maksimum LOB individu adalah 4.096 KB. Jika Anda menggunakan beberapa nilai untuk beberapa statistikMaximum number of tables to load in paralleldanBulk array size, maka jumlah memori yang dibutuhkan untuk tugas Anda adalah sebagai berikut.

$$\text{Required memory} = 2 * 8 * 4,096 * 1,000 = 65,536,000 \text{ KB}$$

Untuk AWS DMS melakukan performa konversi ini secara optimal, CPU harus tersedia ketika konversi terjadi. Memuat beban berlebihan pada CPU dan tidak memiliki cukup sumber daya CPU dapat mengakibatkan migrasi lambat. AWS DMS dapat bersifat CPU-intensif, terutama ketika melakukan migrasi heterogen dan replikasi seperti migrasi dari Oracle ke PostgreSQL. Penggunaan kelas instans replikasi C4 dapat menjadi pilihan yang bagus untuk situasi ini. Untuk informasi selengkapnya, lihat [Memilih instans replikasi AWS DMS yang tepat untuk migrasi Anda](#).

Metrik instans replikasi

Pemantauan instans replikasi mencakup Amazon CloudWatch metrik untuk beberapa statistik berikut:

Metrik	Deskripsi
AvailableMemory	Perkiraan berapa banyak memori yang tersedia untuk mulai aplikasi baru, tanpa bertukar. Untuk informasi lebih lanjut, lihat nilai MemAvailable dalam bagian /proc/memInfo dari Linux man-pages . Unit: Byte

Metrik	Deskripsi
CPUAllocated	<p>Persentase CPU maksimal yang dialokasikan untuk tugas (0 berarti tidak ada batas).</p> <p>AWS DMS memunculkan metrik ini terhadap dimensi gabungan <code>ReplicationInstanceIdentifier</code> dan <code>ReplicationTaskIdentifier</code> di CloudWatch konsol. Gunakan <code>ReplicationInstanceIdentifier</code>, <code>ReplicationTaskIdentifier</code> kategori untuk melihat metrik ini.</p> <p>Unit: Persen</p>
CPUUtilization	<p>Persentase vCPU yang dialokasikan (CPU virtual) saat ini yang digunakan di instans.</p> <p>Unit: Persen</p>
DiskQueueDepth	<p>Jumlah (I/O) permintaan baca/tulis yang belum terselesaikan yang menunggu untuk mengakses disk.</p> <p>Unit: Count (Jumlah)</p>
FreeStorageSpace	<p>Jumlah ruang penyimpanan yang tersedia.</p> <p>Unit: Bit</p>
FreeMemory	<p>Jumlah memori fisik yang tersedia untuk digunakan oleh aplikasi, cache halaman, dan struktur data kernel sendiri. Untuk informasi lebih lanjut, lihat nilai <code>MemFree</code> dalam bagian <code>/proc/memInfo</code> dari Linux man-pages.</p> <p>Unit: Bit</p>
FreeableMemory	<p>Jumlah memori mengakses acak yang tersedia.</p> <p>Unit: Bit</p>

Metrik	Deskripsi
MemoryAllocated	<p>Alokasi maksimum memori untuk tugas (0 berarti tidak ada batas).</p> <p>AWS DMS memunculkan metrik ini terhadap dimensi gabungan <code>ReplicationInstanceIdentifier</code> dan <code>ReplicationTaskIdentifier</code> di CloudWatch konsol. Gunakan <code>ReplicationInstanceIdentifier</code>, <code>ReplicationTaskIdentifier</code> kategori untuk melihat metrik ini.</p> <p>Unit: MiB</p>
WriteIOPS	<p>Jumlah rata-rata operasi I/O tulis disk per detik.</p> <p>Unit: Jumlah/Detik</p>
ReadIOPS	<p>Jumlah rata-rata operasi I/O baca disk per detik.</p> <p>Unit: Jumlah/Detik</p>
WriteThroughput	<p>Jumlah rata-rata byte yang ditulis pada disk per detik.</p> <p>Unit: Byte/Detik</p>
ReadThroughput	<p>Jumlah rata-rata byte untuk baca dari disk per detik.</p> <p>Unit: Byte/Detik</p>
WriteLatency	<p>Jumlah waktu rata-rata yang diambil per operasi I/O (output) disk.</p> <p>Unit: Milidetik</p>
ReadLatency	<p>Jumlah waktu rata-rata yang diambil per operasi I/O (input) disk.</p> <p>Unit: Milidetik</p>
SwapUsage	<p>Jumlah ruang tukar yang digunakan pada instans replikasi.</p> <p>Unit: Bit</p>

Metrik	Deskripsi
NetworkTransmitThroughput	Lalu lintas jaringan keluar (transmit) pada instans replikasi, termasuk lalu lintas basis data pelanggan dan lalu lintas AWS DMS yang digunakan untuk pemantauan dan replikasi. Unit: Byte/Detik
NetworkReceiveThroughput	Lalu lintas jaringan masuk (penerima) pada instans replikasi, termasuk lalu lintas basis data pelanggan dan lalu lintas AWS DMS yang digunakan untuk pemantauan dan replikasi. Unit: Byte/Detik

Metrik tugas replikasi

Pemantauan tugas replikasi termasuk metrik untuk beberapa statistik berikut.

Metrik	Deskripsi
FullLoadThroughputBandwidthTarget	Data keluar ditransmisikan dari beban penuh untuk target di KB per detik.
FullLoadThroughputRowsTarget	Perubahan keluar dari beban penuh untuk target dalam baris per detik.
CDCIncomingChanges	Jumlah total beberapa peristiwa perubahan di point-in-time yang menunggu untuk beberapa sasaran. Perhatikan bahwa ini tidak sama dengan ukuran nilai perubahan transaksi dari titik akhir sumber. Jumlah large metrik ini biasanya menunjukkan AWS DMS tidak dapat menerapkan perubahan yang ditangkap pada waktu yang tepat, sehingga menyebabkan latensi target tinggi.
CDCChangeMemorySource	Jumlah baris yang terakumulasi dalam memori dan menunggu untuk melakukan dari sumber tersebut. Anda dapat melihat metrik ini bersama dengan beberapa metrikChangesDiskSource.

Metrik	Deskripsi
CDCChange sMemoryTarget	Jumlah baris yang terakumulasi dalam memori dan menunggu untuk melakukan dari target tersebut. Anda dapat melihat metrik ini bersama dengan beberapa metrikChangesDiskTarget.
CDCChangesDiskSource	Jumlah baris yang terakumulasi pada disk dan menunggu untuk melakukan dari sumber tersebut. Anda dapat melihat metrik ini bersama dengan beberapa metrikChangesMemorySource.
CDCChangesDiskTarget	Jumlah baris yang terakumulasi dalam memori dan menunggu untuk melakukan dari target tersebut. Anda dapat melihat metrik ini bersama dengan beberapa metrikChangesMemoryTarget.
CDCThroughputBandwidthTarget	Data keluar ditransmisikan untuk target di KB per detik. CDCThroughputBandwidth mencatat data keluar yang dikirimkan pada beberapa titik pengambilan sampel. Jika tidak ada lalu lintas jaringan tugas yang ditemukan, nilai tersebut adalah nol. Karena CDC tidak mengeluarkan transaksi yang berlangsung lama, lalu lintas jaringan mungkin tidak memiliki catatan.
CDCThroughputRowsSource	Perubahan tugas masuk dari sumber dalam baris per detik.
CDCThroughputRowsTarget	Perubahan tugas keluar untuk target dalam baris per detik.

Metrik	Deskripsi
CDCLatencySource	<p>Selang waktu, dalam detik, antara tindakan terakhir yang ditangkap dari titik akhir sumber dan stempel waktu sistem saat ini dari instans AWS DMS. CDCLatencySource mewakili latensi antara sumber dan beberapa contoh replikasi. CDC tinggiLatencySource berarti proses menangkap beberapa perubahan dari beberapa sumber tertunda. Untuk mengidentifikasi latensi dalam replikasi yang sedang berlangsung, Anda dapat melihat metrik iniLatencyTarget. Jika kedua CDCLatencySource dan CDCLatencyTarget tinggi, selidiki CDCLatencySourcePertama.</p> <p>CDCSourceLatency bisa 0 ketika tidak ada jeda replikasi antara sumber dan instance replikasi. CDCSourceLatencyjuga bisa menjadi nol ketika tugas replikasi mencoba membaca peristiwa berikutnya di log transaksi sumber dan tidak ada peristiwa baru dibandingkan dengan terakhir kali dibaca dari sumber. Ketika ini terjadi, tugas me-reset CDCSourceLatency ke 0.</p>
CDCLatencyTarget	<p>Kesenjangan, dalam detik, antara tempel waktu tindakan pertama yang menunggu untuk melakukan pada target dan stempel waktu saat ini dari instans AWS DMS. Latensi target adalah perbedaan antara waktu server instance replikasi dan id peristiwa tertua yang belum dikonfirmasi yang diteruskan ke komponen target. Dengan kata lain, latensi target adalah perbedaan stempel waktu antara instance replikasi dan peristiwa tertua yang diterapkan tetapi belum dikonfirmasi oleh titik akhir TRG (99%). Ketika CDCLatencyTarget tinggi, ini menunjukkan proses penerapan peristiwa untuk target tertunda. Untuk mengidentifikasi latensi dalam replikasi yang sedang berlangsung, Anda dapat melihat metrik iniLatencySource. Jika CDCLatencyTarget tinggi tapi CDCLatencySource tidak tinggi, selidiki jika:</p> <ul style="list-style-type: none">• Tidak ada kunci primer atau indeks yang berada di target• Sumber daya kemacetan terjadi dalam instans target atau replikasi• Masalah jaringan berada di antara instans replikasi dan target

Metrik	Deskripsi
CPUUtilization	<p>Persentase CPU yang digunakan oleh tugas di beberapa inti. Semantik CPUUtilization tugas sedikit berbeda dari CPUUtilization replikasi. Jika 1 vCPU sepenuhnya digunakan, itu menunjukkan 100%, tetapi jika beberapa vCPU sedang digunakan, nilai tersebut bisa di atas 100%.</p> <p>Unit: Persen</p>
SwapUsage	<p>Jumlah pertukaran yang digunakan oleh tugas.</p> <p>Unit: Bit</p>
MemoryUsage	<p>Kelompok kontrol (cgroup)memori.usage_in_bytesDikonsumsi oleh suatu tugas. DMS menggunakan cgroups untuk mengontrol penggunaan sumber daya sistem seperti memori dan CPU. Metrik ini menunjukkan penggunaan memori tugas dalam Megabyte dalam cgroup yang dialokasikan untuk tugas itu. Batas cgroup didasarkan pada sumber daya yang tersedia untuk kelas instance replikasi DMS Anda.memori.usage_in_bytesterdiri dari residen set size (RSS), cache, dan komponen swap memori. Sistem operasi dapat merebut kembali memori cache jika diperlukan. Kami menyarankan Anda untuk beberapa metrik replikasi untuk beberapa metrik replikasi,Available Memory.</p> <p>AWS DMSmemunculkan metrik ini terhadap dimensi gabunganReplicationInstanceIdentifer danReplicationTaskIdentifer di CloudWatch konsol. GunakanReplicationInstanceIdentifer, ReplicationTaskIdentifer kategori untuk melihat metrik ini.</p>

Melihat dan mengelola AWSMencatat tugas DMS

Anda dapat menggunakan Amazon CloudWatch untuk mencatat informasi tugas selamaAWS DMSproses migrasi Anda mengaktifkan untuk mencatat pada pilihan pengaturan tugas. Untuk informasi selengkapnya, lihat [Pengaturan tugas pengelolan](#).

Untuk melihat informasi yang mencatat tugas yang berjalan, ikuti langkah berikut:

1. Membuka konsol AWS DMS tersebut, memilih Tugas migrasi basis data dari panel navigasi. Dialog tugas migrasi basis data muncul.
2. Tentukan pilihan nama tugas Anda. Dialog detail Gambaran Umum muncul.
3. Temukan log tugas migrasi bagian dan pilih Lihat CloudWatch Log.

Anda dapat menggunakan AWS CLI atau AWS DMS API untuk melihat informasi yang mencatat tugas. Untuk melakukannya, gunakan perintah `describe-replication-instance-task-logs` AWS CLI atau AWS DMS tindakan API `DescribeReplicationInstanceTaskLogs`.

Misalnya, perintah AWS CLI berikut ini menunjukkan metadata yang mencatat tugas dalam format JSON.

```
$ aws dms describe-replication-instance-task-logs \
  --replication-instance-arn arn:aws:dms:us-east-1:237565436:rep:CDSFSFSFFFSSUFCAIY
```

Respons sampel dari perintah tersebut adalah sebagai berikut.

```
{
  "ReplicationInstanceTaskLogs": [
    {
      "ReplicationTaskArn": "arn:aws:dms:us-
east-1:237565436:task:MY34U6Z4MSY52GRTIX304AY",
      "ReplicationTaskName": "mysql-to-ddb",
      "ReplicationInstanceTaskLogSize": 3726134
    }
  ],
  "ReplicationInstanceArn": "arn:aws:dms:us-east-1:237565436:rep:CDSFSFSFFFSSUFCAIY"
}
```

Dalam respon ini, ada yang mencatat tugas tunggal (`mysql-to-ddb`) yang terkait dengan instans replikasi. Mencatat pengukuran ini yaitu 3,726,124 byte.

Anda dapat menggunakan informasi yang dikembalikan oleh `describe-replication-instance-task-logs` untuk mendiagnosis dan memecahkan masalah dengan mencatat tugas. Sebagai contoh, jika Anda mengaktifkan untuk mencatat debug detail untuk suatu tugas, dengan

mencatat tugas akan berkembang cepat—berpotensi menghabiskan semua penyimpanan yang tersedia pada instans replikasi, dan menyebabkan status instans berubah ke `storage-full`. Melalui penjelasan dengan mencatat tugas, Anda dapat menentukan mana yang tidak lagi Anda butuhkan; maka Anda dapat menghapusnya, sehingga membebaskan ruang penyimpanan.

Untuk menghapus catatan tugas untuk suatu tugas, tetapkan pengaturan tugas `DeleteTaskLogs` ke `BETUL`. Sebagai contoh, JSON berikut menghapus catatan tugas ketika memodifikasi suatu tugas menggunakan perintah AWS CLI `modify-replication-task` atau tindakan AWS DMS API `ModifyReplicationTask`.

```
{
  "Logging": {
    "DeleteTaskLogs": true
  }
}
```

Mencatat panggilan API AWS DMS dengan AWS CloudTrail

AWS DMS terintegrasi dengan AWS CloudTrail layanan yang memberikan catatan tindakan yang dilakukan oleh pengguna, peran, AWS layanan di AWS DMS. CloudTrail menangkap semua panggilan API untuk AWS DMS sebagai beberapa peristiwa, termasuk beberapa statistik AWS DMS konsol dan dari panggilan kode ke AWS DMS Operasi API. Jika Anda membuat jejak, Anda dapat mengaktifkan beberapa statistik CloudTrail peristiwa ke beberapa bucket Amazon S3, termasuk beberapa acara untuk beberapa statistik AWS DMS. Jika Anda ingin membuat jejak, Anda dapat melihat peristiwa terbaru CloudTrail konsol di Sejarah acara. Menggunakan informasi yang dikumpulkan oleh CloudTrail Anda dapat menentukan beberapa permintaan untuk beberapa statistik AWS DMS, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan permintaan, dan detail tambahan.

Untuk mempelajari lebih lanjut tentang CloudTrail, lihat [AWS CloudTrail Panduan pengguna](#).

AWS DMS informasi di CloudTrail

CloudTrail diaktifkan pada AWS akun saat Anda membuat akun. Ketika aktivitas terjadi di AWS DMS kegiatan tersebut dicatat dalam CloudTrail acara bersama dengan lainnya AWS Acara layanan di Sejarah acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS Anda. Untuk informasi selengkapnya, lihat [Melihat acara dengan CloudTrail sejarah peristiwa](#).

Untuk catatan berkelanjutan tentang peristiwa di akun AWS Anda, termasuk peristiwa untuk AWS DMS, buat jejak. Jejak memungkinkan CloudTrail untuk beberapa statistik untuk beberapa statistik untuk beberapa bucket Amazon S3. Secara default, saat Anda membuat jejak di dalam konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah AWS di partisi AWS dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengkonfigurasi AWS untuk beberapa statistik untuk beberapa statistik untuk beberapa statistik untuk beberapa statistik CloudTrail log. Untuk informasi selengkapnya, lihat :

- [Gambaran umum untuk membuat jejak](#)
- [CloudTrail Layanan yang didukung dan integrasi](#)
- [Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima CloudTrail log koneksi AWS Daerah dan Menerima CloudTrail beberapa statistik log dari beberapa akun](#)

Semua AWS DMS Tindakan log log log log log log log log log koneksi CloudTrail dan didokumentasikan dalam [AWS Database Migration Service Referensi API](#). Misalnya, panggilan ke `CreateReplicationInstance`, `TestConnection` dan `StartReplicationTask` tindakan menghasilkan entri di CloudTrail file log log log log log log log log koneksi

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut:

- Jika permintaan tersebut dibuat dengan kredensial pengguna root atau IAM.
- Jika permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna federasi.
- Bahwa permintaan dibuat oleh layanan AWS lain.

Untuk informasi selengkapnya, lihat [Elemen userIdentity CloudTrail](#) .

Memahami entri file log AWS DMS

Jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai berkas log ke bucket Amazon S3 yang Anda tentukan. CloudTrail beberapa statistik termasuk satu atau beberapa entri log. Sebuah peristiwa mewakili permintaan tunggal dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukan jejak tumpukan yang diurutkan dari panggilan API publik, sehingga file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan CloudTrail entri log yang menunjukkan RebootReplicationInstancetindakan.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AKIAIOSFODNN7EXAMPLE:johndoe",
    "arn": "arn:aws:sts::123456789012:assumed-role/admin/johndoe",
    "accountId": "123456789012",
    "accessKeyId": "ASIAYFI33SINAD0JJEZW",
    "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:EX4MBJ2NMRDL3BMAYJ0XUGYPUE"
  },
  "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:EX4MBJ2NMRDL3BMAYJ0XUGYPUE",
"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"  
}
```

AWS DMSEndit log koneksi

AWS DMS menggunakan pencatatan konteks untuk memberi Anda informasi tentang migrasi yang sedang berlangsung. Pencatatan konteks menulis informasi, seperti berikut ini, ke tugas CloudWatch log:

- Informasi tentang koneksi tugas ke basis data sumber dan target.
- Perilaku tugas replikasi. Anda dapat menggunakan log tugas untuk mendiagnosis masalah replikasi.
- Pernyataan SQL tanpa data yang AWS DMS eksekusi pada beberapa basis data sumber dan target. Anda dapat menggunakan log SQL untuk mendiagnosis perilaku migrasi yang tidak terduga.
- Detail posisi streaming untuk setiap acara CDC.

Pencatatan konteks hanya tersedia di AWS DMS versi 3.5.0 atau lebih tinggi

AWS DMS mengaktifkan pencatatan konteks secara default. Untuk mengontrol pencatatan konteks, atur `EnableLogContext` pengaturan tugas untuk `true` atau `false`, atau dengan memodifikasi tugas di konsol.

AWS DMS menulis informasi log konteks ke CloudWatch tugas replikasi log setiap tiga menit. Pastikan instance replikasi Anda memiliki ruang yang cukup untuk log aplikasinya. Untuk beberapa statistik tentang mengelola log tugas, lihat [Melihat dan mengelola AWS Mencatat tugas DMS](#).

Topik

- [Jenis Objek](#)
- [Contoh Pencatatan](#)
- [Keterbatasan:](#)

Jenis Objek

AWS DMS menghasilkan pencatatan konteks CloudWatch untuk jenis objek berikut.

Jenis Objek	Deskripsi
TABLE_NAME	Entri log ini berisi informasi tentang tabel yang berada dalam cakupan dengan aturan pemetaan tugas saat ini. Anda dapat menggunakan entri ini untuk memeriksa peristiwa tabel untuk periode tertentu selama migrasi.
SCHEMA_NAME	Entri log ini berisi informasi tentang skema yang digunakan oleh aturan pemetaan tugas saat ini. Anda dapat menggunakan entri ini untuk menentukan skema AWS DMS digunakan untuk periode tertentu selama migrasi.
TRANSACTION_ID	Entri ini berisi ID transaksi untuk setiap perubahan DML/DDDL yang diambil dari database sumber. Anda dapat menggunakan entri log ini untuk menentukan perubahan apa yang terjadi selama transaksi tertentu.
CONNECTION_ID	Entri ini berisi ID koneksi. Anda dapat menggunakan entri log ini untuk menentukan koneksi AWS DMS digunakan untuk setiap langkah migrasi.
STATEMENT	Entri ini berisi kode SQL yang digunakan untuk mengambil, memproses, dan menerapkan setiap perubahan migrasi.
STREAM_POSITION	Entri ini berisi posisi dalam file log transaksi untuk setiap tindakan migrasi pada database sumber. Format untuk entri ini bervariasi antara jenis mesin basis data sumber. Anda juga dapat menggunakan informasi ini untuk menentukan posisi awal pos pemeriksaan

Jenis Objek	Deskripsi
	pemulihan saat mengonfigurasi replikasi khusus CDC.

Contoh Pencatatan

Bagian ini berisi contoh catatan log yang dapat Anda gunakan untuk memantau replikasi dan mendiagnosis masalah replikasi.

Contoh log koneksi

Bagian ini berisi sampel log yang menyertakan ID koneksi.

```
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)
```

Contoh log perilaku tugas

Bagian ini berisi sampel log tentang perilaku log tugas replikasi. Anda dapat menggunakan informasi ini untuk mendiagnosis masalah replikasi, misalnya IDLE status.

Berikut ini SOURCE_CAPTURE log menunjukkan bahwa tidak ada peristiwa yang tersedia untuk dibaca dari file log database sumber, dan berisi TARGET_APPLY catatan yang menunjukkan bahwa tidak ada peristiwa yang diterima AWS DMS Komponen CDC untuk beberapa statistik untuk beberapa statistik. Peristiwa ini juga berisi detail konteks terkait peristiwa yang diterapkan sebelumnya.

```
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
(?,?,?,?/?),
```

Contoh log pernyataan SQL

Bagian ini berisi sampel log tentang pernyataan SQL yang dijalankan pada basis data sumber dan target. Pernyataan SQL yang Anda lihat di log hanya menampilkan pernyataan SQL; mereka tidak menampilkan data. Berikut ini TARGET_APPLY log menunjukkan INSERT pernyataan yang berjalan pada target.

```
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
(?,?,?,?/?),
```

Keterbatasan:

Batasan berikut berlaku untuk AWS DMS log konteks koneksi

- Sementara AWS DMS membuat logging minimal untuk semua tipe titik akhir, logging konteks khusus mesin ekstensif hanya tersedia untuk tipe titik akhir berikut. Sebaiknya aktifkan pencatatan konteks saat menggunakan tipe titik akhir ini.
 - MySQL
 - PostgreSQL
 - Oracle
 - Microsoft SQL Server
 - MongoDB/Amazon DocumentDB
 - Amazon S3

Bekerja dengan Amazon EventBridge tindakan dan notifikasi diAWS Database Migration Service

Anda dapat menggunakan Amazon EventBridge untuk memberikan pemberitahuan kapanAWS DMS tindakan terjadi, misalnya penciptaan atau penghapusan instans replikasi. EventBridge menerima pemberitahuan peristiwa dan rute dari suatu peristiwa sebagaimana didefinisikan oleh aturan acara. Anda dapat bekerja dengan notifikasi dalam tindakan yang didukung oleh Amazon EventBridge untuk sebuahAWS Wilayah. Untuk informasi lebih lanjut tentang menggunakan Amazon EventBridge, lihat[Apa Amazon? EventBridge?](#)diAmazon EventBridge Panduan pengguna.

Note

Bekerja dengan Amazon EventBridge acara didukung diAWS DMSversi 3.4.5 dan lebih tinggi.

EventBridge menerima suatu peristiwa, indikator perubahanAWS DMS lingkungan, dan menerapkan aturan untuk merutekan acara ke mekanisme notifikasi. Aturan mencocokkan peristiwa dengan mekanisme pemberitahuan berdasarkan struktur peristiwa, yang disebut pola tindakan.

AWS DMS mengelompokkan peristiwa ke dalam kategori yang dapat Anda terapkan aturan kejadian, sehingga Anda dapat diberi notifikasi peristiwa dalam kategori tersebut. Misalnya, misalnya Anda menerapkan tindakan. EventBridge aturan event ke kategori Creation untuk instance replikasi tertentu. Anda kemudian diberi tahu setiap peristiwa terkait pembuatan yang memengaruhi instans replikasi Anda. Jika Anda menerapkan aturan ke kategori Perubahan untuk instans replikasi, Anda akan diberi tahu saat konfigurasi instans replikasi diubah. Untuk daftar tindakan yang diberikan olehAWS DMS, lihatAWS DMS kategori tindakan dan tindakan, berikut.

Note

Untuk mengizinkan penerbitan dari events.amazonaws.com, pastikan untuk memperbarui kebijakan akses topik Amazon SNS Anda. Untuk informasi selengkapnya, lihat[menggunakan kebijakan berbasis sumber daya untuk Amazon EventBridge](#)diAmazon EventBridge Panduan pengguna.

Untuk informasi selengkapnya tentang memindahkan langganan acara ke Amazon EventBridge, lihat[memigrasikan langganan acara aktif dari DMS ke Amazon EventBridge](#).

Untuk informasi selengkapnya tentang menggunakan tindakan dengan Amazon SNS, lihat [Mengirim dan menerima notifikasi melalui Amazon SNS S S](#).

menggunakan Amazon EventBridge aturan acara untuk AWS DMS

Amazon EventBridge mengirimkan notifikasi peristiwa ke alamat yang Anda berikan saat Anda membuat EventBridge aturan tindakan. Anda mungkin ingin membuat beberapa aturan berbeda. Anda mungkin membuat satu aturan yang menerima semua notifikasi peristiwa dan aturan lain yang hanya mencakup peristiwa penting untuk sumber daya DMS produksi Anda. Anda juga dapat mengaktifkan atau menonaktifkan notifikasi acara di EventBridge.

Untuk membuat Amazon EventBridge aturan yang bereaksi terhadap AWS DMS secara

- Lakukan langkah-langkah yang dijelaskan di [menciptakan Amazon EventBridge aturan yang bereaksi terhadap tindakan](#) di Amazon EventBridge Panduan pengguna, dan menciptakan aturan untuk AWS DMS peristiwa:
 - a. Tentukan tindakan notifikasi ketika tindakan EventBridge menerima tindakan. Ketika sebuah acara cocok, EventBridge mengirim tindakan dan memanggil tindakan.
 - b. Untuk Penyedia layanan, pilih AWS.
 - c. Untuk Nama layanan, pilih Database Migration Service (DMS).

Anda kemudian dapat mulai menerima notifikasi.

Contoh JSON berikut menunjukkan EventBridge model acara untuk AWS DMS layanan.

```
{
  "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:AAAABBBB0CCCCDDDEEEEEE1FFFFF2GGG3FFFFFFF3"
  ],
  "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"
  }
}

```

Untuk daftar semua kategori dan tindakan yang dapat Anda beritahukan, lihat bagian berikut.

AWS DMS kategori tindakan dan tindakan

AWS DMS menghasilkan sejumlah peristiwa dalam kategori yang dapat Anda identifikasi. Setiap kategori berlaku untuk instans replikasi atau jenis sumber tugas replikasi.

Topik

- [ReplicationInstance tindakan](#)
- [ReplicationTask tindakan](#)
- [Pesan tindakan replikasi](#)

ReplicationInstance tindakan

Tabel berikut menunjukkan tindakan dan tindakan untuk tindakan.ReplicationInstance jenis sumber.

Kategori	ID Peristiwa	Deskripsi
Pembuatan	DMS-EVENT-0067	Instans replikasi sedang dibuat.
Penghapusan	DMS-EVENT-0066	Instans replikasi sedang dihapus.
Perubahan Konfigurasi	DMS-EVENT-0012	Kelas instans replikasi untuk instans replikasi ini sedang diubah.
Perubahan Konfigurasi	DMS-EVENT-0018	Penyimpanan untuk instans replikasi sedang ditingkatkan.

Kategori	ID Peristiwa	Deskripsi
Perubahan Konfigurasi	DMS-EVENT-0024	Instans replikasi bertransisi ke konfigurasi Multi-AZ.
Perubahan Konfigurasi	DMS-EVENT-0030	Instans replikasi bertransisi ke konfigurasi Single-AZ.
Pemeliharaan	DMS-EVENT-0026	Pemeliharaan offline untuk instans replikasi sedang berlangsung. Instans replikasi saat ini tidak tersedia.
Pembuatan	DMS-EVENT-0005	instans replikasi.
Penghapusan	DMS-EVENT-0003	instans replikasi telah dihapus.
Perubahan Konfigurasi	DMS-EVENT-0014	Kelas instans replikasi untuk instans replikasi ini sudah berubah.
Perubahan Konfigurasi	DMS-EVENT-0017	Penyimpanan untuk instans replikasi sudah ditingkatkan.
Perubahan Konfigurasi	DMS-EVENT-0025	Instans replikasi telah selesai melakukan transisi ke konfigurasi Multi-AZ.
Perubahan Konfigurasi	DMS-EVENT-0029	Instans replikasi sudah bertransisi ke konfigurasi Single-AZ.
Pemeliharaan	DMS-EVENT-0047	Perangkat lunak pengelolaan pada instans replikasi telah diperbarui.
Pemeliharaan	DMS-EVENT-0027	Pemeliharaan offline untuk instans replikasi selesai. Instans replikasi sekarang tersedia.
Pemeliharaan	DMS-EVENT-0068	instans replikasi dalam tindakan yang tidak dapat ditingkatkan.

Kategori	ID Peristiwa	Deskripsi
Failover	DMS-EVENT-0034	Jika Anda terlalu sering meminta Failover, peristiwa ini terjadi dan bukan peristiwa failover biasa.
Gagal	DMS-EVENT-0031	Contoh replikasi dimasukkan ke dalam keadaan %s.
Kegagalan	DMS-EVENT-0036	Instans replikasi telah gagal karena jaringan yang tidak kompatibel.
Gagal	DMS-EVENT-0037	Ketika layanan tidak dapat mengakses kunci KMS yang digunakan untuk mengenkripsi volume data.
Kegagalan		instans replikasi dimasukkan ke dalam tindakan
Failover		Habis waktu menunggu status aman untuk memulai failover yang diminta pengguna
Failover	DMS-EVENT-0013	Failover mulai untuk instans replikasi Multi-AZ.
Failover	DMS-EVENT-0049	Failover telah selesai untuk instans replikasi.
Failover	DMS-EVENT-0050	Aktivasi multi-AZ telah dimulai.
Failover	DMS-EVENT-0051	Aktivasi multi-AZ selesai.
StateChange		Log kueri umum dan lambat telah diputar secara otomatis sebagai %s

Kategori	ID Peristiwa	Deskripsi
StateChange		AWS DMStidak dapat mengakses kunci enkripsi KMS untuk contoh aplikasi %s. Ini kemungkinan karena kunci dinonaktifkan atauAWS DMStidak dapat mengaksesnya. Jika ini berlanjut , aplikasi akan ditempatkan ke dalam keadaan tidak dapat diakses. Silakan merujuk ke bagian pemecahan masalah diAWS DMSdokumentasi untuk rincian lebih lanjut.
StateChange		AWS DMSsekarang dapat berhasil mengakses kunci enkripsi KMS untuk contoh aplikasi %s.
StateChange		Amazon DMS tidak dapat mengakses kunci enkripsi KMS untuk contoh aplikasi %s. Aplikasi ini akan ditempatkan dalam keadaan tidak dapat diakses. Silakan merujuk ke bagian pemecahan masalah di dokumentasi Amazon DMS untuk detail lebih lanjut.
StateChange		Restart Aplikasi di HM sebagai bagian dari pembuatan Instans Replikasi
StateChange		App Shutdown pada HM sebagai bagian dari penghapusan Instans Replikasi
Failover	DMS-EVENT-0015	Failover multi-AZ untuk tindakan.
LowStorage	DMS-EVENT-0007	Penyimpanan gratis untuk instans replikasi rendah.

Kategori	ID Peristiwa	Deskripsi
LowStorage		Inode yang dialokasikan telah habis - skala penyimpanan untuk diselesaikan

ReplicationTask tindakan

Tabel berikut menunjukkan tindakan dan tindakan untuk tindakan.ReplicationTaskjenis sumber.

Kategori	ID Peristiwa	Deskripsi
Kegagalan	DMS-EVENT-0078	Tugas replikasi terjadi.
Kegagalan	DMS-EVENT-0082	Panggilan untuk membersihkan data tugas gagal.
Perubahan Status	DMS-EVENT-0081	Muat ulang detail tabel telah diminta.
Perubahan Status		Tugas replikasi telah disalin.
Perubahan Status		Salinan tugas replikasi telah gagal.
Perubahan Status		Tugas replikasi telah dipindahkan.
Perubahan Status		Langkah replikasi telah gagal.
Perubahan Status		penciptaan tugas target gagal.
Perubahan Status		tindakan replikasi tindakan telah dimulai.

Kategori	ID Peristiwa	Deskripsi
Perubahan Status		Jalankan penilaian tugas replikasi telah selesai dengan sukses.
Perubahan Status		Jalankan penilaian tugas replikasi telah selesai dengan kegagalan.
StateChange		Jalankan penilaian tugas replikasi telah selesai dengan peringatan.
StateChange		Jalankan penilaian tugas replikasi telah selesai dengan kesalahan.
StateChange		Penilaian tugas replikasi yang dijalankan %s telah dibatalkan.
StateChange		Penilaian tugas replikasi run %s telah dihapus.
StateChange		Jalankan penilaian tugas replikasi gagal menyediakan sumber daya.
StateChange		Tugas replikasi terjadi.
Pembuatan		Tugas replikasi telah terjadi.
ConfigurationChange		Tugas replikasi telah dimodifikasi.
Kegagalan		Tugas replikasi terjadi.
StateChange	DMS-EVENT-0091	Bacaan berhenti, batas tukar file tercapai.
StateChange	DMS-EVENT-0092	Bacaan berhenti, batas penggunaan disk tercapai.

Kategori	ID Peristiwa	Deskripsi
StateChange	DMS-EVENT-0093	Bacaan berhenti, batas penggunaan disk tercapai.
StateChange	DMS-EVENT-0093	Bacaan dilanjutkan.
StateChange	DMS-EVENT-0069	Tugas replikasi telah dimulai dengan TaskType: %s, startType: %s
StateChange	DMS-EVENT-0079	Tugas replikasi dihentikan
Penghapusan	DMS-EVENT-0073	Tugas replikasi telah dihapus.

Pesan tindakan replikasi

Tabel berikut menunjukkan tindakan dan tindakan untuk tindakan.Replikasijenis sumber.

Kategori	Deskripsi
Perubahan Status	Acara penskalaan replikasi DMS.
Perubahan Status	Acara penskalaan replikasi DMS.
Perubahan Status	Acara penskalaan replikasi DMS selesai.
Perubahan Status	replikasi DMS telah terjadi.
Perubahan Status	Replikasi DMS sedang menginisialisasi.
Perubahan Status	Replikasi DMS sedang mempersiapkan sumber daya untuk pengumpulan metadata.
Perubahan Status	Koneksi yang terkait dengan replikasi DMS sedang diuji.
Perubahan Status	Replikasi DMS mengambil metadata

Kategori	Deskripsi
Perubahan Status	Replikasi DMS adalah menghitung kapasitas
Perubahan Status	Replikasi DMS menyediakan kapasitasnya
Perubahan Status	Replikasi DMS telah disediakan.
Perubahan Status	Replikasi DMS telah dimulai
Perubahan Status	Replikasi DMS sedang berjalan.
Perubahan Status	replikasi DMS sedang dihentikan.
Perubahan Status	Replikasi DMS telah berhenti
Perubahan Status	replikasi DMS sedang dimodifikasi.
Perubahan Status	replikasi DMS sedang dihapus.
Perubahan Status	Replikasi DMS adalah deprovisioning kapasitasnya
Perubahan Status	Replikasi DMS telah dideprovisioned.
Kegagalan	replikasi DMS terjadi.

Bekerja dengan acara Amazon SNS dan pemberitahuan diAWS Database Migration Service

Dimulai dengan rilisAWS DMS3.4.5 dan dengan versi yang lebih baru, kami sarankan Anda menggunakan AmazonEventBridgeuntuk memberikan pemberitahuan saatAWS DMSperistiwa terjadi. Untuk informasi lebih lanjut tentang penggunaanEventBridgeacara denganAWS DMS, lihat[Bekerja dengan Amazon EventBridge tindakan dan notifikasi diAWS Database Migration Service](#).

Memindahkan langganan acara ke AmazonEventBridge

Anda dapat menggunakan berikut iniAWS CLIperintah untuk memigrasi langganan acara aktif dari DMS ke AmazonEventBridge, hingga 10 sekaligus.

```
update-subscriptions-to-event-bridge [--force-move | --no-force-move]
```

Secara default,AWS DMShanya memigrasikan langganan acara aktif saat instance replikasi Anda saat iniAWS DMS3.4.5 dan lebih tinggi. Untuk mengganti perilaku default ini, gunakan--force-movepilihan. Namun, beberapa jenis peristiwa mungkin tidak tersedia dengan menggunakan AmazonEventBridgejika contoh replikasi Anda tidak ditingkatkan.

Untuk menjalankanupdate-subscriptions-to-event-bridgePerintah CLI, sebuahAWS Identity and Access ManagementPengguna (IAM) harus memiliki izin kebijakan berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:SetTopicAttributes",
        "events:PutTargets",
        "events:EnableRule",
        "events:PutRule"
      ],
      "Resource": "*"
    }
  ]
}
```

Untuk informasi lebih lanjut tentang memindahkan langganan ke EventBridge, lihat [UpdateSubscriptionsToEventBridge](#) di dalam AWS Database Migration Service Referensi API.

Bekerja dengan acara dan notifikasi Amazon SNS

AWS DMS versi 3.4.5 dan dukungan sebelumnya bekerja dengan peristiwa dan pemberitahuan seperti yang dijelaskan berikut.

AWS Layanan Migrasi Database (AWS DMS) dapat menggunakan Amazon Simple Notification Service (Amazon SNS) untuk memberikan notifikasi saat AWS Acara DMS terjadi, misalnya penciptaan atau penghapusan contoh replikasi. Anda dapat bekerja dengan notifikasi ini dalam bentuk apa pun yang didukung oleh Amazon SNS untuk Wilayah AWS, seperti pesan email, pesan teks, atau panggilan ke titik akhir HTTP.

AWS DMS mengelompokkan tindakan-tindakan ini ke dalam beberapa kategori supaya Anda dapat berlangganan sehingga Anda dapat menerima pemberitahuan saat suatu tindakan dalam kategori tersebut terjadi. Misalnya, jika Anda berlangganan kategori Penciptaan untuk instans replikasi tertentu, Anda akan diberi tahu setiap kali ada tindakan terkait penciptaan yang memengaruhi instans replikasi. Jika Anda berlangganan kategori perubahan konfigurasi untuk instans replikasi, Anda akan diberi tahu saat konfigurasi instans replikasi diubah. Anda juga menerima notifikasi saat langganan notifikasi tindakan berubah. Untuk daftar kategori tindakan yang disediakan oleh AWS DMS, lihat [AWS Kategori acara DMS dan pesan acara untuk pemberitahuan SNS](#), berikut.

AWS DMS mengirim notifikasi tindakan ke alamat yang Anda berikan saat membuat langganan suatu tindakan. Anda mungkin ingin membuat beberapa langganan yang berbeda, seperti satu langganan yang menerima semua notifikasi tindakan dan langganan lain yang hanya mencakup tindakan penting untuk sumber daya DMS produksi Anda. Anda dapat dengan mudah mematikan notifikasi tanpa menghapus langganan dengan membatalkan opsi Mengaktifkan dalam Konsol AWS DMS, atau dengan pengaturan parameter `Enabled` pada SALAH menggunakan DMS API AWS.

Note

Notifikasi tindakan AWS DMS menggunakan pesan teks SMS saat ini tersedia untuk Sumber daya DMS AWS di semua Wilayah AWS yang mendukung Amazon SNS. Untuk daftar Wilayah dan negara AWS tempat Amazon SNS mendukung olahpesan SMS, lihat [Wilayah dan Negara yang Didukung](#).

Untuk informasi selengkapnya tentang penggunaan pesan teks dengan SNS, lihat [Mengirim dan menerima notifikasi SMS menggunakan Amazon SNS](#).

AWS DMS pemberitahuan acara berbeda dari CloudTrail peristiwa di CloudWatch atau EventBridge. CloudTrail pemberitahuan acara dapat dihasilkan oleh pemanggilan API apa pun. DMS mengirimkan notifikasi hanya ketika tindakan DMS terjadi.

AWS DMS menggunakan pengidentifikasi langganan untuk mengidentifikasi setiap langganan. Anda dapat memiliki beberapa langganan tindakan AWS DMS yang diterbitkan untuk topik Amazon SNS yang sama. Bila Anda menggunakan notifikasi tindakan, biaya Amazon SNS berlaku; untuk informasi selengkapnya tentang penagihan Amazon SNS, lihat [Harga Amazon SNS](#).

Untuk berlangganan AWS Peristiwa DMS dengan Amazon SNS, gunakan proses berikut:

1. Buatlah topik Amazon SNS. Dalam topik ini, Anda menentukan jenis notifikasi yang ingin Anda terima dan ke alamat atau nomor yang manakah notifikasi ini akan ditujukan.
2. Buatlah langganan notifikasi tindakan AWS DMS dengan menggunakan API AWS Management Console, AWS CLI, atau AWS DMS.
3. AWS DMS mengirimkan email persetujuan atau pesan SMS ke alamat mana Anda mengirimkan langganan tersebut. Untuk mengonfirmasi langganan Anda, klik tautan di email persetujuan atau pesan SMS.
4. Setelah mengonfirmasi langganan, status langganan Anda diperbarui di bagian Langganan tindakan pada konsol AWS DMS.
5. Anda kemudian mulai menerima notifikasi kejadian.

Untuk daftar semua kategori dan tindakan yang dapat Anda beritahukan, lihat bagian berikut. Untuk detail selengkapnya tentang berlangganan dan bekerja dengan langganan tindakan AWS DMS, lihat [Berlangganan AWS Pemberitahuan acara DMS menggunakan SNS](#).

AWS Kategori acara DMS dan pesan acara untuk pemberitahuan SNS

Important

Dimulai dengan rilis AWS DMS 3.4.5 dan dengan versi yang lebih baru, kami sarankan Anda menggunakan Amazon EventBridge untuk memberikan notifikasi saat AWS DMS peristiwa

terjadi. Untuk informasi lebih lanjut tentang penggunaan EventBridge acara dengan AWS DMS, lihat [Bekerja dengan Amazon EventBridge tindakan dan notifikasi di AWS Database Migration Service](#).

AWS DMS menghasilkan sejumlah besar tindakan dalam beberapa kategori supaya Anda dapat berlangganan dengan menggunakan konsol AWS DMS atau AWS DMS API. Setiap kategori berlaku untuk tipe sumber; saat ini AWS DMS memberi support instans replikasi dan tipe sumber tugas replikasi.

Tabel berikut menunjukkan kemungkinan kategori dan tindakan untuk tipe sumber instans replikasi.

Kategori	DMS event ID	Deskripsi
Perubahan Konfigurasi	DMS-EVENT-0012	Kelas instans replikasi untuk instans replikasi ini sedang diubah.
Perubahan Konfigurasi	DMS-EVENT-0014	Kelas instans replikasi untuk instans replikasi ini sudah berubah.
Perubahan Konfigurasi	DMS-EVENT-0018	Penyimpanan untuk instans replikasi sedang ditingkatkan.
Perubahan Konfigurasi	DMS-EVENT-0017	Penyimpanan untuk instans replikasi sudah ditingkatkan.
Perubahan Konfigurasi	DMS-EVENT-0024	Instans replikasi bertransisi ke konfigurasi Multi-AZ.
Perubahan Konfigurasi	DMS-EVENT-0025	Instans replikasi selesai bertransisi ke konfigurasi Multi-AZ.
Perubahan Konfigurasi	DMS-EVENT-0030	Instans replikasi bertransisi ke konfigurasi Single-AZ.
Perubahan Konfigurasi	DMS-EVENT-0029	Instans replikasi sudah bertransisi ke konfigurasi Single-AZ.
Pembuatan	DMS-EVENT-0067	Instans replikasi sedang dibuat.

Kategori	DMS event ID	Deskripsi
Pembuatan	DMS-EVENT-0005	Instans replikasi dibuat.
Penghapusan	DMS-EVENT-0066	Instans replikasi sedang dihapus.
Penghapusan	DMS-EVENT-0003	Instans replikasi dihapus.
Pemeliharaan	DMS-EVENT-0047	Perangkat lunak pengelolaan pada instans replikasi telah diperbarui.
Pemeliharaan	DMS-EVENT-0026	Pemeliharaan offline untuk instans replikasi sedang berlangsung. Instans replikasi saat ini tidak tersedia.
Pemeliharaan	DMS-EVENT-0027	Pemeliharaan offline untuk instans replikasi selesai. Instans replikasi sekarang tersedia.
Pemeliharaan	DMS-EVENT-0068	Instans replikasi berada dalam status yang tidak dapat dimutakhirkan.
LowStorage	DMS-EVENT-0007	Contoh replikasi telah mengkonsumsi lebih dari 90% penyimpanan yang dialokasikan. Anda dapat memantau ruang penyimpanan untuk contoh replikasi menggunakan metrik Free Storage Space.
Failover	DMS-EVENT-0013	Failover mulai untuk instans replikasi Multi-AZ.
Failover	DMS-EVENT-0049	Failover mulai untuk instans replikasi Multi-AZ.
Failover	DMS-EVENT-0015	Failover Multi-AZ untuk siaga selesai.
Failover	DMS-EVENT-0050	Aktivasi multi-AZ telah dimulai.
Failover	DMS-EVENT-0051	Aktivasi Multi-AZ telah selesai.
Failover	DMS-EVENT-0034	Jika Anda terlalu sering meminta failover, tindakan ini terjadi dan bukan peristiwa failover biasa.

Kategori	DMS event ID	Deskripsi
Gagal	DMS-EVENT-0031	Instans replikasi telah mengalami gagal penyimpanan.
Gagal	DMS-EVENT-0036	Instans replikasi telah gagal karena jaringan yang tidak kompatibel.
Gagal	DMS-EVENT-0037	Layanan tidak dapat mengakses kunci AWS KMS yang digunakan untuk mengenkripsi volume data.

Tabel berikut menunjukkan kemungkinan kategori dan tindakan untuk tipe sumber tugas replikasi.

Kategori	DMS event ID	Deskripsi
Perubahan Status	DMS-EVENT-0069	Tugas replikasi telah dimulai.
Perubahan Status	DMS-EVENT-0081	Unggah ulang detail tabel telah diminta.
Perubahan Status	DMS-EVENT-0079	Tugas replikasi telah berhenti.
Perubahan Status	DMS-EVENT-0091	Bacaan berhenti, batas tukar file tercapai.
Perubahan Status	DMS-EVENT-0092	Bacaan berhenti, batas penggunaan disk tercapai.
Perubahan Status	DMS-EVENT-0093	Bacaan dilanjutkan.
Gagal	DMS-EVENT-0078	Tugas replikasi telah gagal.
Gagal	DMS-EVENT-0082	Panggilan untuk menghapus tugas telah gagal untuk membersihkan data tugas.

Kategori	DMS event ID	Deskripsi
Perubahan Konfigurasi	DMS-EVENT-0080	Tugas replikasi dimodifikasi.
Penghapusan	DMS-EVENT-0073	Tugas replikasi dihapus.
Pembuatan	DMS-EVENT-0074	Tugas replikasi dihapus.

Contoh berikut menunjukkan langganan tindakan AWS DMS dengan kategori Perubahan Status.

```
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
```

Berlangganan AWS Pemberitahuan acara DMS menggunakan SNS

Important

Dimulai dengan rilis AWS DMS 3.4.5 dan dengan versi yang lebih baru, kami sarankan Anda menggunakan Amazon EventBridge untuk memberikan pemberitahuan saat AWS DMS peristiwa terjadi. Untuk informasi lebih lanjut tentang penggunaan EventBridge cara dengan AWS DMS, lihat [Bekerja dengan Amazon EventBridge tindakan dan notifikasi di AWS Database Migration Service](#).

Anda dapat membuat langganan notifikasi tindakan AWS DMS sehingga Anda dapat diberitahu ketika tindakan AWS DMS terjadi. Cara termudah untuk membuat langganan adalah dengan konsol AWS DMS tersebut. Dalam langganan notifikasi, Anda memilih cara dan tempat untuk mengirim notifikasi. Anda menentukan tipe sumber yang ingin Anda ketahui; saat ini AWS DMS mendukung instans replikasi dan tipe sumber tugas replikasi. Tergantung pada tipe sumber pilihan Anda, Anda

memilih kategori tindakan dan mengidentifikasi sumber untuk menerima notifikasi tindakan yang Anda inginkan.

Menggunakan AWS Management Console

Important

Dimulai dengan rilis AWS DMS 3.4.5 dan dengan versi yang lebih baru, kami sarankan Anda menggunakan Amazon EventBridge untuk memberikan notifikasi saat AWS DMS peristiwa terjadi. Untuk informasi lebih lanjut tentang penggunaan EventBridge secara bersama-sama dengan AWS DMS, lihat [Bekerja dengan Amazon EventBridge tindakan dan notifikasi di AWS Database Migration Service](#).

Untuk berlangganan AWS Pemberitahuan peristiwa DMS dengan Amazon SNS dengan menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.

Jika Anda masuk sebagai pengguna IAM, pastikan Anda memiliki izin yang sesuai untuk mengakses AWS DMS.

2. Di panel navigasi, pilih Berlangganan tindakan.
3. Pada halaman Berlangganan tindakan, pilih Membuat langganan tindakan.
4. Pada halaman Membuat langganan tindakan, melakukan hal berikut:
 - a. Dalam Detail, untuk Nama, memasukkan nama untuk langganan notifikasi tindakan.
 - b. Memilih Mengaktifkan untuk mengaktifkan langganan. Jika Anda ingin berlangganan, tetapi belum ingin mengirim notifikasi, jangan memilih Mengaktifkan.
 - c. Dalam Target, memilih salah satu Topik yang ada, Membuat topik email baru atau Membuat topik SMS baru untuk mengirim notifikasi. Pastikan bahwa Anda memiliki topik Amazon SNS yang ada untuk mengirim notifikasi atau membuat topik. Jika Anda membuat suatu topik, Anda dapat memasukkan alamat email tempat notifikasi akan dikirim.
 - d. Dalam Sumber tindakan, untuk tipe sumber, memilih tipe sumber. Satu-satunya pilihan adalah instans replikasi dan tugas replikasi.
 - e. Tergantung pada tipe sumber pilihan Anda, Anda dapat memilih kategori dan sumber tindakan untuk notifikasi tindakan yang Anda inginkan.

Create event subscription

Details

Name

The name for your event subscription

 Enabled

Target

Send notification to

- Existing topics
- Create new email topic
- Create new SMS topic

Topic name**With these recipients**

Email addresses or phone numbers of SMS enabled devices to send the notifications to

Event source

Source type

Source Type of resource this subscription will consume events from

Event categories

- All event categories
- Select specific event categories

Replication instance

- All instances
- Select specific instances

- f. Pilihan Membuat langganan tindakan.

Konsol AWS DMS menunjukkan bahwa langganan sedang dibuat.

Note

Anda juga dapat membuat langganan pemberitahuan peristiwa Amazon SNS menggunakan AWS DMS API dan CLI. Untuk informasi lebih lanjut, lihat [CreateEventSubscription](#) di dalam AWS DMS Referensi API dan [create-event-subscription](#) di dalam AWS DMS Referensi CLI dokumentasi.

Memvalidasi kebijakan akses topik SNS Anda

Kebijakan akses SNS Anda memerlukan izin yang memungkinkan AWS DMS untuk mempublikasikan acara ke topik SNS Anda. Anda dapat memvalidasi dan memperbarui kebijakan akses seperti yang dijelaskan dalam prosedur berikut.

Untuk memvalidasi kebijakan akses Anda

1. Buka konsol Amazon SNS.
2. Dari panel navigasi, pilih **Topik** dan pilih topik yang ingin Anda terima pemberitahuan DMS.
3. Pilih **Kebijakan akses** tab.

Anda dapat memperbarui kebijakan jika kebijakan akses SNS Anda tidak mengizinkan AWS DMS untuk mempublikasikan acara ke topik SNS Anda.

Untuk memperbarui kebijakan akses Anda

1. Dari **Detail** bagian dari halaman topik Anda, pilih **Mengedit**.
2. Perluas **Kebijakan akses** bagian, dan lampirkan kebijakan berikut ke editor JSON.

```
{
  "Sid": "dms-allow-publish",
  "Effect": "Allow",
  "Principal": {
    "Service": "dms.amazonaws.com"
  },
}
```

```
"Action": "sns:Publish",  
"Resource": "your-SNS-topic-ARN"  
}
```

Kami menyarankan Anda lebih lanjut membatasi akses ke topik SNS Anda dengan menentukan `aws:SourceArn` kondisi, yang merupakan `DMSEventSubscriptionArn` yang menerbitkan acara ke topik.

```
...  
"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. Pilih Simpan perubahan.

Validasi data AWS DMS

Topik

- [Statistik tugas replikasi](#)
- [Statistik tugas replikasi dengan Amazon CloudWatch](#)
- [Memvalidasi ulang tabel selama pengerjaan tugas](#)
- [Menggunakan editor JSON untuk memodifikasi aturan validasi](#)
- [Tugas hanya validasi](#)
- [Pemecahan Masalah](#)
- [Kinerja Validasi Pergeseran Merah](#)
- [Batasan](#)
- [Validasi data target Amazon S3](#)

AWS DMS menyediakan dukungan untuk validasi data untuk memastikan bahwa data Anda dimigrasi secara akurat dari sumber ke target. Jika diaktifkan, validasi dimulai segera setelah beban penuh dilakukan untuk sebuah tabel. Validasi membandingkan perubahan tambahan untuk tugas yang CDC-nya diaktifkan saat terjadi.

Selama validasi data, AWS DMS membandingkan setiap baris dalam sumber dengan baris yang sesuai pada target, memverifikasi baris berisikan data yang sama, dan melaporkan setiap ketidakcocokan. Untuk melakukan hal tersebut AWS DMS mengeluarkan kueri yang tepat untuk mengambil data. Perhatikan bahwa kueri ini akan mengonsumsi sumber daya tambahan pada sumber dan target serta sumber daya jaringan tambahan.

Untuk tugas CDC saja dengan validasi diaktifkan, semua data yang sudah ada sebelumnya dalam tabel divalidasi sebelum memulai validasi data baru.

Validasi data bekerja dengan database sumber berikut di mana pun AWS DMS mendukungnya sebagai titik akhir sumber:

- Oracle
- Database yang kompatibel dengan PostgreSQL (PostgreSQL, Aurora PostgreSQL, atau Aurora Tanpa Server untuk PostgreSQL)
- Database yang kompatibel dengan MySQL (MySQL, MariaDB, Aurora MySQL, atau Aurora Tanpa Server untuk MySQL)

- Microsoft SQL Server
- IBM Db2 LUW

Validasi data bekerja dengan database target berikut di mana pun AWS DMS mendukungnya sebagai titik akhir target:

- Oracle
- Database yang kompatibel dengan PostgreSQL (PostgreSQL, Aurora PostgreSQL, atau Aurora Tanpa Server untuk PostgreSQL)
- Database yang kompatibel dengan MySQL (MySQL, MariaDB, Aurora MySQL, atau Aurora Tanpa Server untuk MySQL)
- Microsoft SQL Server
- IBM Db2 LUW
- Amazon Redshift
- Amazon S3. Untuk informasi tentang memvalidasi data target Amazon S3, lihat [Validasi data target Amazon S3](#)

Untuk informasi lebih lanjut tentang titik akhir yang didukung, lihat [Bekerja dengan Titik akhir DMS AWS](#).

Validasi data memerlukan waktu tambahan, melampaui jumlah yang diperlukan untuk migrasi itu sendiri. Waktu tambahan yang diperlukan tergantung pada berapa banyak data yang dimigrasi.

Untuk informasi selengkapnya tentang pengaturan ini, lihat [Pengaturan tugas validasi data](#).

Untuk contoh pengaturan `ValidationSettings` tugas dalam file JSON, lihat [Contoh pengaturan tugas](#).

Statistik tugas replikasi

Ketika validasi data diaktifkan, AWS DMS menyediakan statistik berikut di tingkat tabel:

- `ValidationState`—Status validasi tabel. Parameter dapat memiliki nilai berikut:
 - `Not enabled`—Validasi tidak diaktifkan untuk tabel dalam tugas migrasi.
 - `Pending records`—Beberapa catatan dalam tabel menunggu validasi.

- **Mismatched records**—Beberapa catatan dalam tabel tidak cocok antara sumber dan target. Ketidakcocokan mungkin terjadi karena sejumlah alasan; Untuk informasi lebih lanjut, periksa tabel `awsdms_control.awsdms_validation_failures_v1` pada titik akhir target.
- **Suspended records**—Beberapa catatan dalam tabel tidak dapat divalidasi.
- **No primary key**—Tabel tidak dapat divalidasi karena tidak memiliki kunci primer.
- **Table error**—Tabel tidak divalidasi karena berada dalam status kesalahan dan beberapa data tidak dimigrasi.
- **Validated**—Semua baris dalam tabel divalidasi. Jika tabel diperbarui, status dapat berubah dari **Validated**.
- **Error**—Tabel tidak dapat divalidasi karena ada kesalahan tidak terduga.
- **Pending validation**—Tabel menunggu validasi.
- **Mempersiapkan tabel**—Mempersiapkan tabel diaktifkan dalam tugas migrasi untuk validasi.
- **Pending revalidation**—Validasi semua baris dalam tabel tertunda setelah tabel diperbarui.
- **ValidationPending**—Jumlah catatan yang telah dimigrasikan ke target, tetapi itu belum divalidasi.
- **ValidationSuspended**—Jumlah catatan yang tidak AWS DMS bisa dibandingkan. Misalnya, jika catatan di sumber terus diperbarui, AWS DMS tidak dapat membandingkan sumber dan target.
- **ValidationFailed**—Jumlah catatan yang tidak lulus fase validasi data.

Untuk contoh pengaturan `ValidationSettings` tugas dalam file JSON, lihat [Contoh pengaturan tugas](#).

Anda dapat melihat informasi validasi data menggunakan konsol, AWS CLI, atau AWS DMS API.

- Di konsol, Anda dapat memilih untuk memvalidasi tugas saat Anda membuat atau memodifikasi tugas. Untuk melihat laporan validasi data menggunakan konsol, pilih tugas di halaman Tugas dan pilih tab Statistik tabel di bagian detail.
- Menggunakan CLI, atur `EnableValidation` parameter `true` saat membuat atau memodifikasi tugas untuk memulai validasi data. Contoh berikut membuat tugas dan mengaktifkan validasi data.

```
create-replication-task
  --replication-task-settings '{"ValidationSettings":{"EnableValidation":true}}'
  --replication-instance-arn arn:aws:dms:us-east-1:5731014:
    rep:36KWMB7Q
  --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"}]}'
```

Gunakan perintah `describe-table-statistics` untuk menerima laporan validasi data dalam format JSON. Perintah berikut menunjukkan laporan validasi data.

```
aws dms describe-table-statistics --replication-task-arn arn:aws:dms:us-
east-1:5731014:
rep:36KWVMB7Q
```

Laporan tersebut akan serupa dengan berikut ini.

```
{
  "ReplicationTaskArn": "arn:aws:dms:us-west-2:5731014:task:VFPPTYKK2RYSI",
  "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
    }
  ]
}
```

- Menggunakan AWS DMS API, buat tugas menggunakan `CreateReplicationTask` tindakan dan atur `EnableValidation` parameter ke `true` untuk memvalidasi data yang dimigrasikan oleh tugas.

Gunakan `DescribeTableStatisticstindakan` untuk menerima laporan validasi data dalam format JSON.

Statistik tugas replikasi dengan Amazon CloudWatch

Saat Amazon CloudWatch diaktifkan, AWS DMS berikan statistik tugas replikasi berikut:

- `ValidationSucceededRecordCount`— Jumlah baris yang AWS DMS divalidasi, per menit.
- `ValidationAttemptedRecordCount`— Jumlah baris yang validasi dicoba, per menit.
- `ValidationFailedOverallCount`— Jumlah baris di mana validasi gagal.
- `ValidationSuspendedOverallCount`— Jumlah baris di mana validasi ditangguhkan.
- `ValidationPendingOverallCount`— Jumlah baris di mana validasi masih tertunda.
- `ValidationBulkQuerySourceLatency`— AWS DMS dapat melakukan validasi data secara massal, terutama dalam skenario tertentu selama replikasi beban penuh atau sedang berlangsung ketika ada banyak perubahan. Metrik ini menunjukkan latensi yang diperlukan untuk membaca kumpulan data massal dari titik akhir sumber.
- `ValidationBulkQueryTargetLatency`— AWS DMS dapat melakukan validasi data secara massal, terutama dalam skenario tertentu selama replikasi beban penuh atau sedang berlangsung ketika ada banyak perubahan. Metrik ini menunjukkan latensi yang diperlukan untuk membaca kumpulan data massal dari titik akhir target.
- `ValidationItemQuerySourceLatency`— Selama replikasi yang sedang berlangsung, validasi data dapat mengidentifikasi perubahan yang sedang berlangsung dan memvalidasi perubahan tersebut. Metrik ini menunjukkan latensi dalam membaca perubahan tersebut dari sumber. Validasi dapat menjalankan lebih banyak kueri dari yang diperlukan, berdasarkan jumlah perubahan, jika ada kesalahan selama validasi.
- `ValidationItemQueryTargetLatency`— Selama replikasi yang sedang berlangsung, validasi data dapat mengidentifikasi perubahan yang sedang berlangsung dan memvalidasi perubahan baris demi baris. Metrik ini memberi kita latensi dalam membaca perubahan tersebut dari target. Validasi dapat menjalankan lebih banyak kueri dari yang diperlukan, berdasarkan jumlah perubahan, jika ada kesalahan selama validasi.

Untuk mengumpulkan informasi validasi data dari statistik yang CloudWatch diaktifkan, pilih Aktifkan CloudWatch log saat Anda membuat atau memodifikasi tugas menggunakan konsol. Kemudian, untuk melihat informasi validasi data dan memastikan bahwa data Anda dimigrasi secara akurat dari sumber ke target, lakukan hal berikut.

1. Pilih tugas di halaman Tugas migrasi basis data.
2. Pilih tab CloudWatch metrik.
3. Pilih Validasi dari menu tarik turun.

Memvalidasi ulang tabel selama pengerjaan tugas

Saat tugas sedang berjalan, Anda dapat meminta AWS DMS untuk melakukan validasi data.

AWS Management Console

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.

Jika Anda masuk sebagai pengguna AWS Identity and Access Management (IAM), pastikan bahwa Anda memiliki izin yang sesuai untuk mengakses AWS DMS. izin yang diperlukan, lihat [Izin IAM diperlukan untuk menggunakan AWS DMS](#).

2. Pilih Tugas dari panel navigasi.
3. Pilih tugas berjalan yang memiliki tabel yang ingin Anda validasi ulang.
4. Pilih tab Statistik Tabel.
5. Pilih tabel yang ingin Anda validasi ulang (Anda dapat memilih hingga 10 tabel sekaligus). Jika tugas tidak lagi berjalan, Anda tidak dapat memvalidasi ulang tabel.
6. Pilih Validasi Ulang.

Menggunakan editor JSON untuk memodifikasi aturan validasi

Untuk menambahkan aturan validasi untuk tugas yang menggunakan editor JSON dari Konsol AWS DMS, lakukan hal berikut:

1. Pilih Tugas migrasi basis data.
2. Pilih tugas Anda dari daftar tugas migrasi.
3. Jika tugas Anda sedang berjalan, pilih Berhenti dari menu tarik turun Tindakan.
4. Setelah tugas telah berhenti, untuk memodifikasi tugas Anda, pilih Modifikasi dari menu tarik turun Tindakan.
5. Di bagian Pemetaan tabel, pilih Editor JSON dan tambahkan aturan validasi Anda ke pemetaan tabel Anda.

Misalnya, Anda dapat menambahkan aturan validasi berikut untuk menjalankan fungsi ganti pada sumber. Dalam kasus ini, jika aturan validasi bertemu null byte, byte tersebut akan divalidasi sebagai spasi.

```
{
  "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}"
}
```

Tugas hanya validasi

Anda hanya dapat membuat tugas validasi untuk melihat pratinjau dan memvalidasi data tanpa melakukan migrasi atau replikasi data apa pun. Untuk membuat tugas validasi saja, atur `ValidationOnly` pengaturan `EnableValidation` dan `true`. Saat mengaktifkan `ValidationOnly`, persyaratan tambahan berlaku. Untuk informasi selengkapnya, lihat [Pengaturan tugas validasi data](#).

Untuk jenis migrasi hanya beban penuh, tugas validasi hanya selesai jauh lebih cepat daripada yang setara dengan CDC ketika banyak kegagalan dilaporkan. Tetapi perubahan pada titik akhir sumber atau target dilaporkan sebagai kegagalan untuk mode beban penuh, kemungkinan kerugian.

Hanya tugas validasi CDC yang menunda validasi berdasarkan latensi rata-rata, dan mencoba ulang kegagalan beberapa kali sebelum melaporkannya. Jika sebagian besar perbandingan data mengakibatkan kegagalan, tugas validasi hanya untuk mode CDC sangat lambat, kelemahan potensial.

Tugas hanya validasi harus diatur dalam arah yang sama dengan tugas replikasi, terutama untuk CDC. Ini karena tugas Hanya Validasi CDC mendeteksi baris mana yang telah berubah dan perlu divalidasi ulang berdasarkan log perubahan pada sumbernya. Jika target ditentukan sebagai sumber,

maka ia hanya tahu tentang perubahan yang dikirim ke target oleh DMS dan tidak dijamin untuk menangkap kesalahan replikasi.

Validasi beban penuh saja

Dimulai dengan AWS DMS versi 3.4.6 dan yang lebih tinggi, tugas validasi beban penuh hanya dengan cepat membandingkan semua baris dari sumber dan tabel target dalam satu lintasan, segera melaporkan kegagalan, dan kemudian dimatikan. Validasi tidak pernah ditangguhkan karena kegagalan dalam mode ini, itu dioptimalkan untuk kecepatan. Tetapi perubahan pada titik akhir sumber atau target dilaporkan sebagai kegagalan.

Note

Dimulai dengan AWS DMS versi 3.4.6 dan yang lebih tinggi, perilaku validasi ini juga berlaku untuk tugas migrasi beban penuh dengan validasi diaktifkan.

Hanya validasi CDC

Tugas validasi CDC hanya memvalidasi semua baris yang ada antara tabel sumber dan target pada awal yang baru. Selain itu, tugas validasi CDC hanya berjalan terus menerus, memvalidasi ulang perubahan replikasi yang sedang berlangsung, membatasi jumlah kegagalan yang dilaporkan setiap pass, dan mencoba ulang baris yang tidak cocok sebelum gagal. Ini dioptimalkan untuk mencegah positif palsu.

Validasi untuk tabel (atau seluruh tugas) ditangguhkan jika `TableFailureMaxCount` ambang batas `FailureMaxCount` atau dilanggar. Ini juga berlaku untuk tugas migrasi CDC atau Full Load+CDC dengan validasi diaktifkan. Dan tugas CDC dengan validasi mengaktifkan penundaan validasi ulang untuk setiap baris yang diubah berdasarkan latensi sumber dan target rata-rata.

Tetapi tugas validasi CDC saja tidak memigrasikan data dan tidak memiliki latensi. Ini diatur `ValidationQueryCdcDelaySeconds` ke 180 secara default. Dan Anda dapat meningkatkan jumlah untuk memperhitungkan lingkungan latensi tinggi dan membantu mencegah positif palsu.

Validasi hanya kasus penggunaan

Kasus penggunaan untuk memisahkan bagian validasi data dari tugas migrasi atau replikasi menjadi tugas validasi terpisah yang hanya mencakup, tetapi tidak terbatas pada, hal-hal berikut:

- Kontrol tepat saat validasi terjadi - Kueri validasi menambahkan beban tambahan ke titik akhir sumber dan target. Jadi, memigrasi atau mereplikasi data dalam satu tugas terlebih dahulu, kemudian memvalidasi hasil di tugas lain dapat bermanfaat.
- Kurangi beban pada instance replikasi — Memisahkan validasi data untuk dijalankan pada instansinya sendiri dapat menguntungkan.
- Dapatkan dengan cepat berapa banyak baris yang tidak cocok pada saat tertentu — Misalnya, tepat sebelum atau selama pemotongan produksi jendela pemeliharaan — ke titik akhir target, Anda dapat membuat tugas validasi Full Load saja untuk mendapatkan jawaban atas pertanyaan Anda.
- Ketika kegagalan validasi diharapkan untuk tugas migrasi dengan komponen CDC — Misalnya, jika memigrasikan Oracle `varchar2` ke PostgreSQL, validasi CDC terus mencoba ulang `jsonb` baris gagal ini dan membatasi jumlah kegagalan yang dilaporkan setiap kali. Namun, Anda dapat membuat tugas validasi Full Load saja dan mendapatkan jawaban yang lebih cepat.
- Anda telah mengembangkan skrip/utilitas perbaikan data yang membaca tabel kegagalan validasi — (Lihat juga,). [Pemecahan Masalah](#) Tugas validasi Full Load hanya dengan cepat melaporkan kegagalan untuk skrip perbaikan data untuk ditindaklanjuti.

Untuk contoh pengaturan `ValidationSettings` tugas dalam file JSON, lihat [Contoh pengaturan tugas](#)).

Pemecahan Masalah

Selama validasi, AWS DMS menciptakan tabel baru di titik akhir target:

`awsdms_control.aws_dms_validation_failures_v1`. Jika ada catatan yang memasuki `ValidationSuspended` atau `ValidationFailed` negara bagian, AWS DMS tulis informasi diagnostik ke `awsdms_control.aws_dms_validation_failures_v1`. Anda dapat mengkueri tabel ini untuk membantu memecahkan kesalahan validasi.

Untuk informasi tentang mengubah skema default tabel dibuat pada target, lihat [Mengontrol pengaturan tugas tabel](#).

Berikut ini adalah deskripsi dari tabel `awsdms_control.aws_dms_validation_failures_v1`:

Nama kolom	Tipe data	Deskripsi
TASK_NAME	VARCHAR(128) NOT NULL	Pengidentifikasi tugas AWS DMS.
TABLE_OWNER	VARCHAR(128) NOT NULL	Skema (pemilik) dari tabel.
TABLE_NAME	VARCHAR(128) NOT NULL	Nama tabel.
FAILURE_TIME	DATETIME(3) NOT NULL	Waktu ketika kegagalan terjadi.
KEY_TYPE	VARCHAR(128) NOT NULL	Dicadangkan untuk penggunaan di masa mendatang (nilai selalu 'Row')
KEY	TEXT NOT NULL	Ini adalah kunci primer untuk jenis catatan baris.
FAILURE_TYPE	VARCHAR(128) NOT NULL	Tingkat kepelikan kesalahan validasi. Bisa RECORD_DIFF , MISSING_SOURCE atau MISSING_TARGET .
DETAILS	VARCHAR(8000) NOT NULL	String yang diformat JSON dari semua nilai kolom sumber/target yang tidak cocok untuk kunci tertentu.

Kueri berikut akan menunjukkan kepada Anda semua kegagalan untuk sebuah tugas dengan mengkueri tabel `awsdms_validation_failures_v1`. Nama tugas harus ID sumber daya eksternal tugas tersebut. ID sumber daya eksternal dari tugas adalah nilai terakhir dalam ARN tugas. Misalnya, untuk tugas dengan nilai ARN `arn:aws:dms:us-west-2:5599:task:VFPFKH4FJR3FTYKK2RYSI`, ID sumber daya eksternal tugas tersebut adalah `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'}],]
```

Anda dapat melihat bidang DETAILS untuk menentukan kolom yang tidak cocok. Karena Anda memiliki kunci primer dari catatan yang gagal, Anda dapat mengkueri titik akhir sumber dan target untuk melihat bagian dari catatan yang tidak cocok.

Kinerja Validasi Pergeseran Merah

Amazon Redshift berbeda dari database relasional dalam beberapa hal, termasuk penyimpanan kolumnar, MPP, kompresi data, dan faktor lainnya. Perbedaan ini memberikan Redshift profil kinerja yang berbeda dari database relasional.

Selama fase replikasi beban penuh, validasi menggunakan kueri rentang, dengan ukuran data diatur oleh pengaturan. `PartitionSize` Kueri berbasis rentang ini memilih semua catatan dari tabel sumber.

Untuk replikasi yang sedang berlangsung, kueri beralih antara pengambilan berdasarkan rentang dan rekaman individual. Jenis kueri ditentukan secara dinamis berdasarkan beberapa faktor, seperti berikut ini:

- Volume kueri
- Jenis kueri DML pada tabel sumber
- Latensi tugas
- Jumlah total catatan
- Pengaturan validasi seperti `PartitionSize`

Anda mungkin melihat beban tambahan di kluster Amazon Redshift karena kueri validasi. Karena faktor-faktor di atas bervariasi di seluruh kasus penggunaan, Anda harus meninjau kinerja kueri validasi Anda dan menyetel kluster dan tabel Anda sesuai dengan itu. Beberapa opsi untuk mengurangi ises kinerja meliputi:

- Kurangi ThreadCount pengaturan PartitionSize dan untuk membantu mengurangi beban kerja selama validasi beban penuh. Perhatikan bahwa ini akan memperlambat validasi data.
- Meskipun Redshift tidak menerapkan kunci utama, AWS DMS bergantung pada kunci utama untuk mengidentifikasi catatan secara unik pada target untuk validasi data. Jika memungkinkan, setel kunci utama untuk mencerminkan kunci pengurutan sehingga kueri validasi beban penuh dijalankan lebih cepat.

Batasan

- Validasi data mengharuskan tabel memiliki kunci primer atau indeks unik.
 - Kolom kunci primer tidak dapat berupa jenis CLOB, BLOB, atau BYTE.
 - Untuk kolom kunci primer jenis VARCHAR atau CHAR, panjangnya harus kurang dari 1024. Anda harus menentukan panjang dalam tipe data. Anda tidak dapat menggunakan tipe data tak terbatas sebagai kunci utama untuk validasi data.
 - Kunci Oracle yang dibuat dengan NOVALIDATE klausa tidak dianggap sebagai kunci utama atau indeks unik.
 - Untuk tabel Oracle tanpa kunci utama dan hanya kunci unik, kolom dengan kendala unik juga harus memiliki kendala. NOT NULL
- Validasi nilai NULL PK/UK tidak didukung.
- Jika kolasi kolom kunci primer dalam instans PostgreSQL target tidak diatur ke "C", urutan kunci primer berbeda dibandingkan dengan urutan di Oracle. Jika urutan berbeda antara PostgreSQL dan Oracle, validasi data gagal untuk memvalidasi catatan.
- Validasi data menghasilkan kueri tambahan terhadap basis data sumber dan target. Anda harus memastikan bahwa kedua basis data memiliki cukup sumber daya untuk menangani beban tambahan ini. Ini terutama berlaku untuk target Redshift. Untuk informasi lebih lanjut, lihat [Kinerja Validasi Pergeseran Merah](#) berikut ini.
- Validasi data tidak didukung saat mengkonsolidasikan beberapa database menjadi satu.
- Untuk titik akhir Oracle sumber atau target, AWS DMS menggunakan DBMS_CRYPTO untuk memvalidasi LOB. Jika titik akhir Oracle Anda menggunakan LOB, maka Anda harus memberikan izin eksekusi pada dbms_crypto ke akun pengguna yang digunakan untuk mengakses titik akhir Oracle. Anda dapat melakukan hal tersebut dengan menjalankan pernyataan berikut:

```
grant execute on sys.dbms_crypto to dms_endpoint_user;
```

- Jika basis data target dimodifikasi di luar AWS DMS selama validasi, maka diskrepansi mungkin tidak dilaporkan secara akurat. Hasil ini dapat terjadi jika salah satu aplikasi Anda menulis data ke tabel target, sementara AWS DMS sedang melakukan validasi pada tabel yang sama.
- Jika satu atau beberapa baris terus dimodifikasi selama validasi, maka tidak AWS DMS dapat memvalidasi baris tersebut.
- Jika AWS DMS mendeteksi lebih dari 10.000 catatan yang gagal atau ditangguhkan, validasi akan dihentikan. Sebelum Anda melanjutkan lebih jauh, selesaikan masalah yang mendasari dengan data tersebut.
- AWS DMS tidak mendukung validasi data tampilan.
- AWS DMS tidak mendukung validasi data saat pengaturan tugas substitusi karakter digunakan.
- AWS DMS tidak mendukung validasi tipe Oracle LONG.
- AWS DMS tidak mendukung validasi tipe Oracle Spatial selama migrasi heterogen.

Untuk batasan saat menggunakan validasi target S3, lihat [Batasan untuk menggunakan validasi target S3](#)

Validasi data target Amazon S3

AWS DMS mendukung memvalidasi data yang direplikasi di target Amazon S3. Karena AWS DMS menyimpan data yang direplikasi sebagai file datar di Amazon S3, kami menggunakan kueri [Amazon CREATE TABLE AS SELECT Athena](#) (CTAS) untuk memvalidasi data.

Kueri pada data yang disimpan di Amazon S3 sangat intens secara komputasi. Dengan demikian, AWS DMS menjalankan validasi pada data Amazon S3 selama pengambilan data perubahan (CDC) hanya sekali sehari, pada tengah malam (00:00) UTC. Setiap validasi harian yang AWS DMS berjalan disebut validasi interval. Selama validasi interval, AWS DMS validasi semua catatan perubahan yang dimigrasikan ke bucket Amazon S3 target selama 24 jam sebelumnya. Untuk informasi selengkapnya tentang batasan validasi interval, lihat [Batasan untuk menggunakan validasi target S3](#).

Validasi target Amazon S3 menggunakan Amazon Athena, jadi biaya tambahan berlaku. Untuk informasi selengkapnya, lihat [Harga Amazon Athena](#).

Note

Validasi target S3 membutuhkan AWS DMS versi 3.5.0 atau yang lebih baru.

Topik

- [Prasyarat validasi target S3](#)
- [Izin untuk menggunakan validasi target S3](#)
- [Batasan untuk menggunakan validasi target S3](#)
- [Mengggunakan tugas validasi saja dengan validasi target S3](#)

Prasyarat validasi target S3

Sebelum menggunakan validasi target S3, periksa pengaturan dan izin berikut:

- Tetapkan DataFormat nilai untuk [S3Settings](#) endpoint Anda ke. `parquet` Untuk informasi selengkapnya, lihat [Pengaturan parket untuk S3](#).
- Pastikan peran yang ditetapkan ke akun pengguna yang digunakan untuk membuat tugas migrasi memiliki kumpulan izin yang benar. Lihat [Izin](#) berikut.

Untuk tugas yang menggunakan replikasi berkelanjutan (CDC), periksa pengaturan berikut:

- Aktifkan pencatatan tambahan sehingga Anda memiliki catatan lengkap dalam data CDC. Untuk informasi tentang mengaktifkan logging tambahan, lihat [Secara otomatis menambahkan supplemental logging untuk titik akhir sumber Oracle](#) di [Pemecahan masalah dan dukungan diagnostik](#) bagian dalam panduan ini.
- Tetapkan `TimestampColumnName` parameter untuk titik akhir target. Tidak ada batasan pada nama kolom stempel waktu. Untuk informasi selengkapnya, lihat [S3Settings](#).
- Siapkan partisi folder berbasis tanggal untuk target. Untuk informasi selengkapnya, lihat [Mengggunakan partisi folder berdasarkan tanggal](#).

Izin untuk menggunakan validasi target S3

Untuk menyiapkan akses untuk menggunakan validasi target S3, pastikan bahwa peran yang ditetapkan ke akun pengguna yang digunakan untuk membuat tugas migrasi memiliki kumpulan izin berikut. Ganti nilai sampel dengan nilai Anda.

```
{
  "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>/*"
    ]
  }
]

```

```
    }  
  ]  
}
```

Batasan untuk menggunakan validasi target S3

Lihat batasan tambahan berikut yang berlaku saat menggunakan validasi target S3. Untuk batasan yang berlaku untuk semua validasi, lihat [Batasan](#)

- `DatePartitionSequence` Nilai Anda membutuhkan komponen Hari. Validasi target S3 tidak mendukung format. `YYYYMM`
- Ketika validasi interval berjalan selama CDC, Anda mungkin melihat kesalahan validasi palsu dalam tabel. `awsdms_validation_failures_v1` Kesalahan ini terjadi karena AWS DMS memigrasikan perubahan yang tiba selama validasi interval ke folder partisi hari berikutnya. Biasanya, perubahan ini ditulis ke dalam folder partisi hari ini. Kesalahan palsu ini adalah batasan memvalidasi replikasi dari database sumber dinamis ke target statis, seperti Amazon S3. Untuk menyelidiki kesalahan palsu ini, periksa catatan di dekat akhir jendela validasi (00:00 UTC), yaitu saat kesalahan ini biasanya muncul.

Untuk meminimalkan jumlah kesalahan palsu, pastikan bahwa `CDCLatencySource` untuk tugas rendah. Untuk informasi tentang pemantauan latensi, lihat [Metrik tugas replikasi](#).

- Tugas di `stopped` negara bagian `failed` atau tidak memvalidasi perubahan hari sebelumnya. Untuk meminimalkan kesalahan validasi karena kegagalan yang tidak terduga, buat tugas validasi terpisah saja dengan pemetaan tabel yang sama, serta titik akhir sumber dan target. Untuk informasi selengkapnya tentang tugas validasi saja, lihat [Menggunakan tugas validasi saja dengan validasi target S3](#).
- Kolom Status Validasi dalam statistik tabel mencerminkan keadaan validasi interval terbaru. Akibatnya, tabel yang memiliki ketidakcocokan mungkin muncul sebagai divalidasi setelah validasi interval hari berikutnya. Periksa bucket Amazon S3 target untuk ketidakcocokan yang terjadi lebih dari sehari yang lalu. `s3_validation_failures` folder
- Validasi S3 menggunakan fitur tabel bucketed dari Amazon Athena. Hal ini memungkinkan validasi S3 untuk membuat salinan berember dari data tabel target. Ini berarti bahwa salinan data tabel dibagi menjadi subset yang cocok dengan partisi internal validasi DMS. Meja berember Athena memiliki batas 100.000 ember. Setiap tabel yang coba divalidasi oleh validasi S3 yang melebihi

batas ini akan gagal validasi. Jumlah bucket yang coba dibuat oleh Validasi S3 sama dengan yang berikut:

```
(#records in the table) / (validation partition size setting)
```

Untuk mengatasi batasan ini, tingkatkan pengaturan ukuran partisi validasi sehingga jumlah bucket yang dibuat oleh Validasi S3 kurang dari 100.000. Untuk informasi selengkapnya tentang bucketing, lihat [Partisi dan bucketing di Athena di Panduan Pengguna Amazon Athena](#).

Menggunakan tugas validasi saja dengan validasi target S3

Tugas hanya validasi menjalankan validasi pada data yang akan dimigrasi tanpa menjalankan migrasi.

Hanya tugas validasi yang terus berjalan, meskipun tugas migrasi berhenti, yang memastikan bahwa AWS DMS tidak melewatkan jendela validasi interval 00:00 UTC.

Menggunakan tugas hanya validasi dengan titik akhir target Amazon S3 memiliki batasan berikut:

- Validasi Amazon S3 untuk Tugas Beban Penuh dengan setelan Hanya Validasi diaktifkan didukung, tetapi beroperasi secara berbeda dari tugas Beban Penuh dan Hanya Validasi untuk titik akhir lainnya. Untuk S3 sebagai Target, tugas jenis ini memvalidasi hanya terhadap Data Beban Penuh di target S3, dan tidak akan memvalidasi terhadap data apa pun yang dimigrasi sebagai bagian dari migrasi CDC. Hanya gunakan fitur ini untuk memvalidasi data yang dibuat oleh tugas Full-Load saja. Menggunakan mode ini untuk memvalidasi data dalam target yang menjalankan tugas CDC aktif tidak akan menghasilkan validasi yang efektif.
- Hanya tugas validasi hanya memvalidasi perubahan sejak jendela validasi interval terakhir (00:00 UTC). Hanya tugas validasi yang tidak memvalidasi data muatan penuh atau data CDC dari hari-hari sebelumnya.

Penandaan sumber daya di AWS Database Migration Service

Anda dapat menggunakan tag di AWS Database Migration Service (AWS DMS) untuk menambahkan metadata ke sumber daya Anda. Anda juga dapat menggunakan tag ini dengan kebijakan AWS Identity and Access Management (IAM) untuk mengelola akses ke sumber daya AWS DMS dan untuk mengontrol tindakan yang dapat diterapkan ke sumber daya AWS DMS. Terakhir, Anda dapat menggunakan tag ini untuk melacak biaya dengan mengelompokkan pengeluaran untuk sumber daya serupa yang ditandai.

Semua sumber daya AWS DMS dapat ditandai:

- Sertifikat
- Penyedia data
- Migrasi data
- Titik akhir
- Langganan kejadian
- Profil instans
- Proyek migrasi
- Instans replikasi
- Kelompok subnet replikasi
- Tugas replikasi

Tag AWS DMS adalah pasangan nama dan nilai yang Anda tentukan dan tautkan dengan sumber daya AWS DMS. Nama ini disebut sebagai kunci. Memberikan nilai untuk kunci bersifat opsional. Anda dapat menggunakan tag untuk memberikan informasi tambahan ke sumber daya AWS DMS. Kunci tag dapat digunakan, misalnya, untuk menentukan kategori, dan nilai tag dapat berupa item dalam kategori tersebut. Misalnya, Anda dapat menentukan kunci tag "proyek" dan nilai tag "Salix", yang menunjukkan bahwa sumber daya AWS DMS ditugaskan ke proyek Salix. Anda juga dapat menggunakan tag untuk menunjuk sumber daya AWS DMS seperti yang digunakan untuk tes atau produksi dengan menggunakan kunci seperti lingkungan=tes atau lingkungan =produksi. Kami merekomendasikan bahwa Anda menggunakan serangkaian kunci tag yang konsisten guna mempermudah pelacakan metadata yang terkait dengan sumber daya AWS DMS.

Gunakan tag untuk mengelola tagihan AWS Anda agar sesuai dengan struktur biaya Anda. Untuk melakukannya, daftar untuk mendapatkan tagihan Akun AWS Anda dengan menyertakan nilai kunci tanda. Lalu, untuk melihat biaya sumber daya gabungan, organisasikan informasi penagihan Anda sesuai dengan sumber daya Anda dengan nilai kunci tanda yang sama. Misalnya, Anda dapat memberi tag pada beberapa sumber daya dengan nama aplikasi tertentu, kemudian mengelola informasi penagihan Anda untuk melihat total biaya aplikasi tersebut di beberapa layanan. Untuk informasi lebih lanjut, lihat [Penggunaan Tag Alokasi Biaya](#) dalam Panduan Pengguna AWS Billing.

Setiap sumber daya AWS DMS memiliki rangkaian tag, yang berisi semua tag yang ditetapkan ke sumber daya AWS DMS tersebut. Rangkaian tag dapat berisi sepuluh tag atau kosong. Jika Anda menambahkan tag ke sumber daya AWS DMS yang memiliki kunci yang sama dengan tag yang ada pada sumber daya, nilai yang baru akan menimpa nilai yang lama.

AWS tidak menerapkan makna semantik pada tag Anda; tag ditafsirkan secara ketat sebagai string karakter. AWS DMS mungkin mengatur tag pada sumber daya AWS DMS, bergantung pada pengaturan yang Anda gunakan saat membuat sumber daya.

Daftar berikut menjelaskan karakteristik tag DMS AWS.

- Kunci tag adalah nama tag yang diperlukan. Nilai string dapat terdiri dari 1 hingga 128 karakter Unicode dan tidak boleh diawali dengan "aws:" atau "dms:". String hanya dapat berisi kumpulan huruf Unicode, angka, spasi, '_', '.', '/', '=', '+', '-' (Java regex: "`^([\\p{L}\\p{Z}\\p{N}_. :/=+\\ \\-]*)$`").
- Nilai tag adalah nilai string opsional dari tag. Nilai string dapat terdiri dari 1 hingga 256 karakter Unicode dan tidak boleh diawali dengan "aws:" atau "dms:". String hanya dapat berisi kumpulan huruf Unicode, angka, spasi, '_', '.', '/', '=', '+', '-' (Java regex: "`^([\\p{L}\\p{Z}\\p{N}_. :/=+\\ \\-]*)$`").

Nilai rangkaian tag tidak harus unik dan bisa nol. Misalnya, Anda dapat menggunakan pasangan nilai kunci dalam satu rangkaian tag `project/Trinity` dan `cost-center/Trinity`.

Anda dapat menggunakan AWS CLI atau API AWS DMS untuk menambahkan, mencantumkan, dan menghapus tag pada sumber daya AWS DMS. Saat menggunakan AWS CLI atau API AWS DMS, Anda harus menyediakan Amazon Resource Name (ARN) untuk sumber daya AWS DMS yang ingin Anda gunakan. Untuk informasi lebih lanjut tentang cara membuat ARN, lihat [Membangun Nama Sumber Daya Amazon \(ARN\) untuk AWS DMS](#).

Perhatikan bahwa tag disimpan di cache untuk diotorisasi. Oleh karena itu, penambahan dan pembaruan tag di sumber daya AWS DMS dapat memakan waktu beberapa menit sebelum tag itu tersedia.

API

Anda dapat menambahkan, mencantumkan, atau menghapus tag untuk sumber daya AWS DMS menggunakan API AWS DMS.

- Untuk menambahkan tag ke sumber daya AWS DMS, gunakan [AddTagsToResource](#) operasi.
- Untuk mencantumkan tag yang ditetapkan ke sumber daya AWS DMS, gunakan operasi [ListTagsForResource](#).
- Untuk menghapus tag dari sumber daya AWS DMS, gunakan operasi [RemoveTagsFromResource](#).

Untuk mempelajari selengkapnya tentang cara membuat ARN yang diperlukan, lihat [Membangun Nama Sumber Daya Amazon \(ARN\) untuk AWS DMS](#).

Ketika menggunakan XML dengan API AWS DMS, tag menggunakan skema berikut:

```
<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Trinity</Value>
    </Tag>
    <Tag>
      <Key>User</Key>
      <Value>Jones</Value>
    </Tag>
  </TagSet>
</Tagging>
```

Tabel berikut menyediakan daftar tag XML yang diperbolehkan beserta karakteristiknya. Perhatikan bahwa nilai untuk Kunci dan Nilai bergantung pada huruf besar-kecil. Misalnya, project=Trinity dan PROJECT=Trinity adalah dua tag yang berbeda.

Elemen tag	Deskripsi
TagSet	Rangkaian tag adalah wadah untuk semua tag yang ditetapkan ke sumber daya Amazon RDS. Hanya ada satu rangkaian tag per sumber daya. Anda bekerja dengan TagSet hanya melalui API AWS DMS.
Tanda	Tag adalah pasangan nilai kunci yang ditentukan pengguna. Satu rangkaian tag bisa berisi 1 hingga 10 tag.
Kunci	<p>Kunci adalah nama tag yang diperlukan. Nilai string dapat terdiri dari 1 hingga 128 karakter Unicode dan tidak boleh diawali dengan "dms:" atau "aws:". String hanya dapat berisi kumpulan huruf Unicode, angka, spasi, '_', '!', '/', '=', '+', '-' (Java regex: <code>"^([\p{L}\p{Z}\p{N}_.:/+\\-]*)\$" </code>).</p> <p>Kunci dalam rangkaian tag harus unik. Misalnya, Anda tidak dapat memiliki pasangan kunci dalam satu rangkaian tag dengan kunci yang sama tetapi dengan nilai yang berbeda, seperti project/Trinity dan project/Xanadu.</p>
Nilai	<p>Nilai adalah nilai opsional dari tag. Nilai string dapat terdiri dari 1 hingga 256 karakter Unicode dan tidak boleh diawali dengan "dms:" atau "aws:". String hanya dapat berisi kumpulan huruf Unicode, angka, spasi, '_', '!', '/', '=', '+', '-' (Java regex: <code>"^([\p{L}\p{Z}\p{N}_.:/+\\-]*)\$" </code>).</p> <p>Nilai rangkaian tag tidak harus unik dan bisa nol. Misalnya, Anda dapat menggunakan pasangan nilai kunci dalam satu rangkaian tag project/Trinity dan cost-center/Trinity.</p>

Keamanan di AWS Database Migration Service

Keamanan cloud di AWS merupakan prioritas tertinggi. Sebagai seorang pelanggan AWS, Anda mendapatkan manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan dari organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara AWS dan Anda. [Model tanggung jawab bersama](#) menggambarkan ini sebagai keamanan dari cloud dan keamanan di dalam cloud:

- Keamanan cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di dalam AWS Cloud. AWS juga memberi layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [program kepatuhan AWS](#). Untuk mempelajari program kepatuhan yang berlaku di AWS DMS, lihat [Cakupan layanan menurut program kepatuhan AWS](#).
- Keamanan dalam cloud – Tanggung jawab Anda ditentukan oleh layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan organisasi Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS DMS. Topik berikut menunjukkan kepada Anda cara mengonfigurasi AWS DMS untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan layanan AWS lain yang membantu Anda memantau dan mengamankan sumber daya AWS DMS Anda.

Anda dapat mengelola akses ke sumber daya AWS DMS dan basis data (DB) Anda. Metode yang Anda gunakan untuk mengelola akses tergantung pada tugas replikasi yang perlu Anda lakukan dengan AWS DMS:

- Gunakan kebijakan AWS Identity and Access Management (IAM) untuk menetapkan izin yang menentukan siapa yang diizinkan mengelola sumber daya AWS DMS. AWS DMS mengharuskan Anda memiliki izin yang sesuai jika Anda masuk sebagai pengguna IAM. Misalnya, Anda dapat menggunakan IAM untuk menentukan siapa yang diperbolehkan untuk membuat, menjelaskan, memodifikasi, dan menghapus instans dan kluster DB, menandai sumber daya, atau memodifikasi grup keamanan. Untuk informasi lebih lanjut tentang IAM dan penggunaannya dengan AWS DMS, lihat [Identitas dan manajemen akses untuk AWS Database Migration Service](#).

- AWS DMS menggunakan Lapisan Soket Aman (SSL) untuk koneksi titik akhir Anda dengan Keamanan Lapisan Pengangkutan (TLS). Untuk informasi lebih lanjut tentang penggunaan SSL/TLS dengan AWS DMS, lihat [Menggunakan SSL dengan AWS Database Migration Service](#).
- AWS DMS menggunakan kunci enkripsi AWS Key Management Service (AWS KMS) untuk mengenkripsi penyimpanan yang digunakan oleh instans replikasi Anda dan informasi koneksi titik akhir. AWS DMS juga menggunakan kunci enkripsi AWS KMS untuk mengamankan data at rest target Anda untuk titik akhir target Amazon S3 dan Amazon Redshift. Untuk informasi selengkapnya, lihat [Mengatur kunci enkripsi dan menentukan izin AWS KMS](#).
- AWS DMS selalu membuat instans replikasi Anda dalam virtual private cloud (VPC) berdasarkan layanan Amazon VPC untuk kendali akses jaringan terbesar yang memungkinkan. Untuk instans DB Anda dan instans kluster, gunakan VPC yang sama dengan instans replikasi Anda, atau VPC tambahan agar sesuai dengan tingkat kontrol akses. Setiap Amazon VPC yang Anda gunakan harus dikaitkan dengan grup keamanan yang memiliki aturan yang mengizinkan semua lalu lintas di semua port untuk meninggalkan (keluar) VPC. Pendekatan ini memungkinkan komunikasi dari instans replikasi ke titik akhir basis data sumber dan target, selama ingress yang benar diaktifkan pada titik akhir tersebut.

Untuk informasi lebih lanjut tentang konfigurasi jaringan yang tersedia untuk AWS DMS, lihat [Menyiapkan jaringan untuk instans replikasi](#). Untuk informasi lebih lanjut tentang pembuatan instans DB atau instans kluster di VPC, lihat dokumentasi manajemen keamanan dan kluster untuk basis data Amazon Anda di [AWS dokumentasi](#). Untuk informasi lebih lanjut tentang konfigurasi jaringan yang didukung AWS DMS, lihat [Menyiapkan jaringan untuk instans replikasi](#).

- Untuk melihat log migrasi database, Anda memerlukan izin CloudWatch Log Amazon yang sesuai untuk peran IAM yang Anda gunakan. Untuk informasi lebih lanjut tentang pencatatan untuk AWS DMS, lihat [Pemantauan tugas replikasi menggunakan Amazon CloudWatch](#).

Topik

- [Perlindungan data di AWS Database Migration Service](#)
- [Identitas dan manajemen akses untuk AWS Database Migration Service](#)
- [Validasi kepatuhan untuk AWS Database Migration Service](#)
- [Ketahanan di AWS Database Migration Service](#)
- [Keamanan infrastruktur dalam AWS Database Migration Service](#)
- [Kontrol akses detail menggunakan nama sumber daya dan tag](#)
- [Mengatur kunci enkripsi dan menentukan izin AWS KMS](#)

- [Keamanan jaringan untuk AWS Database Migration Service](#)
- [Menggunakan SSL dengan AWS Database Migration Service](#)
- [Penggubahan kata sandi basis data](#)

Perlindungan data di AWS Database Migration Service

Enkripsi data

Anda dapat mengaktifkan enkripsi untuk sumber daya data titik akhir AWS DMS yang didukung. AWS DMS juga mengenkripsi koneksi ke AWS DMS dan di antara AWS DMS dan semua titik akhir sumber dan targetnya. Selain itu, Anda dapat mengelola kunci yang digunakan oleh AWS DMS dan titik akhir targetnya yang didukung untuk mengaktifkan enkripsi ini.

Topik

- [Enkripsi saat istirahat](#)
- [Enkripsi dalam transit](#)
- [Manajemen kunci](#)

Enkripsi saat istirahat

AWS DMS mendukung enkripsi saat istirahat dengan mengizinkan Anda untuk menentukan mode enkripsi sisi server yang ingin Anda gunakan untuk mendorong data yang direplikasi ke Amazon S3 sebelum disalin ke titik akhir target AWS DMS yang didukung. Anda dapat menentukan mode enkripsi ini dengan mengatur `encryptionMode` atribut koneksi tambahan untuk titik akhir. Jika `encryptionMode` pengaturan ini menentukan mode enkripsi kunci KMS, Anda juga dapat membuat AWS KMS kunci khusus khusus untuk mengenkripsi data target untuk titik akhir target berikut: AWS DMS

- Amazon Redshift - Untuk informasi lebih lanjut tentang pengaturan `encryptionMode`, lihat [Pengaturan titik akhir saat menggunakan Amazon Redshift sebagai target untuk AWS DMS](#). Untuk informasi selengkapnya tentang membuat kunci AWS KMS enkripsi kustom, lihat [Membuat dan menggunakan kunci AWS KMS untuk mengenkripsi data target Amazon Redshift](#).
- Amazon S3 – Untuk informasi lebih lanjut tentang pengaturan `encryptionMode`, lihat [Pengaturan titik akhir saat menggunakan Amazon S3 sebagai target untuk AWS DMS](#). Untuk informasi selengkapnya tentang membuat kunci AWS KMS enkripsi kustom, lihat [Membuat kunci AWS KMS untuk mengenkripsi objek target Amazon S3](#).

Enkripsi dalam transit

AWS DMS mendukung enkripsi dalam transit dengan memastikan bahwa data yang direplikasi bergerak dengan aman dari titik akhir sumber ke titik akhir target. Ini termasuk mengenkripsi bucket S3 pada instans replikasi yang digunakan oleh tugas replikasi Anda untuk penyimpanan menengah saat data bergerak melalui alur replikasi. Untuk mengenkripsi koneksi tugas ke sumber dan titik akhir target AWS DMS menggunakan Secure Socket Layer (SSL) atau Transport Layer Security (TLS). Dengan mengenkripsi koneksi ke kedua titik akhir, AWS DMS memastikan bahwa data Anda aman saat bergerak baik dari titik akhir sumber ke tugas replikasi Anda dan dari tugas Anda ke titik akhir target. Untuk informasi lebih lanjut tentang penggunaan SSL/TLS dengan AWS DMS, lihat [Menggunakan SSL dengan AWS Database Migration Service](#)

AWS DMS mendukung kunci default dan kustom untuk mengenkripsi penyimpanan replikasi menengah dan informasi koneksi. Anda mengelola kunci ini dengan menggunakan AWS KMS. Untuk informasi selengkapnya, lihat [Mengatur kunci enkripsi dan menentukan izin AWS KMS](#).

Manajemen kunci

AWS DMS mendukung kunci default atau kustom untuk mengenkripsi penyimpanan replikasi, informasi koneksi, dan penyimpanan data target untuk titik akhir target tertentu. Anda mengelola kunci ini dengan menggunakan AWS KMS. Untuk informasi selengkapnya, lihat [Mengatur kunci enkripsi dan menentukan izin AWS KMS](#).

Privasi lalu lintas kerja internet

Koneksi disediakan dengan perlindungan antara AWS DMS dan titik akhir sumber dan target dalam Wilayah AWS yang sama, baik berjalan secara on premise atau sebagai bagian dari layanan AWS di cloud. (Setidaknya satu titik akhir, sumber atau target, harus dijalankan sebagai bagian dari layanan AWS di cloud.) Perlindungan ini berlaku jika komponen tersebut berbagi virtual private cloud (VPC) yang sama atau ada di VPC terpisah, jika semua VPC tersebut berada di Wilayah AWS yang sama. Untuk informasi lebih lanjut tentang konfigurasi jaringan yang didukung untuk AWS DMS, lihat [Menyiapkan jaringan untuk instans replikasi](#). Untuk informasi lebih lanjut tentang pertimbangan keamanan saat menggunakan konfigurasi jaringan ini, lihat [Keamanan jaringan untuk AWS Database Migration Service](#).

Perlindungan data di DMS Fleet Advisor

DMS Fleet Advisor mengumpulkan dan menganalisis metadata database Anda untuk menentukan ukuran target migrasi yang tepat. DMS Fleet Advisor tidak mengakses data di tabel Anda dan tidak

mentransfernya. Selain itu, DMS Fleet Advisor tidak melacak penggunaan fitur database dan tidak mengakses statistik penggunaan Anda.

Anda mengontrol akses ke database Anda ketika Anda membuat pengguna database yang digunakan DMS Fleet Advisor untuk bekerja dengan database Anda. Anda memberikan hak istimewa yang diperlukan kepada pengguna ini. Untuk menggunakan DMS Fleet Advisor, Anda memberi pengguna database Anda izin baca. DMS Fleet Advisor tidak memodifikasi database Anda dan tidak memerlukan izin menulis. Untuk informasi selengkapnya, lihat [Membuat pengguna database untuk AWS DMS Fleet Advisor](#).

Anda dapat menggunakan enkripsi data di database Anda. AWS DMS juga mengenkripsi koneksi dalam DMS Fleet Advisor dan dalam pengumpul datanya.

Pengumpul data DMS menggunakan antarmuka pemrograman aplikasi Perlindungan Data (DPAPI) untuk mengenkripsi, melindungi, dan menyimpan informasi tentang lingkungan pelanggan dan kredensial basis data. DMS Fleet Advisor menyimpan data terenkripsi ini dalam file di server tempat pengumpul data DMS Anda bekerja. DMS Fleet Advisor tidak mentransfer data ini dari server ini. Untuk informasi selengkapnya tentang DPAPI, lihat [Cara: Menggunakan Perlindungan Data](#).

Setelah Anda menginstal pengumpul data DMS, Anda dapat melihat semua kueri yang dijalankan aplikasi ini untuk mengumpulkan metrik. Anda dapat menjalankan pengumpul data DMS dalam mode offline dan kemudian meninjau data yang dikumpulkan di server Anda. Selain itu, Anda dapat meninjau data yang dikumpulkan ini di bucket Amazon S3 Anda. Untuk informasi selengkapnya, lihat [Bagaimana cara kerja pengumpul data DMS?](#).

Identitas dan manajemen akses untuk AWS Database Migration Service

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. AWS DMS IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Bagaimana AWS Database Migration Service bekerja dengan IAM](#)
- [AWS Database Migration Service contoh kebijakan berbasis identitas](#)
- [Contoh kebijakan berbasis sumber daya untuk AWS KMS](#)
- [Menggunakan rahasia untuk mengakses titik akhir AWS Database Migration Service](#)
- [Menggunakan peran terkait layanan untuk AWS DMS](#)
- [Memecahkan masalah AWS Database Migration Service identitas dan akses](#)
- [Izin IAM diperlukan untuk menggunakan AWS DMS](#)
- [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#)
- [Pencegahan confused deputy lintas layanan](#)
- [Kebijakan terkelola AWS untuk AWS Database Migration Service](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan. AWS DMS

Pengguna layanan — Jika Anda menggunakan AWS DMS layanan untuk melakukan pekerjaan Anda, maka administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak AWS DMS fitur untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara mengelola akses dapat membantu Anda meminta

izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di AWS DMS, lihat [Memecahkan masalah AWS Database Migration Service identitas dan akses](#).

Administrator layanan — Jika Anda bertanggung jawab atas AWS DMS sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS DMS. Tugas Anda adalah menentukan AWS DMS fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM AWS DMS, lihat [Bagaimana AWS Database Migration Service bekerja dengan IAM](#).

Administrator IAM – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih detail tentang cara Anda menulis kebijakan untuk mengelola akses ke AWS DMS. Untuk melihat contoh kebijakan AWS DMS berbasis identitas yang dapat Anda gunakan di IAM, lihat [AWS Database Migration Service contoh kebijakan berbasis identitas](#)

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensial identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensial Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas gabungan, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat [Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS](#).

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani permintaan AWS API](#) di Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari lebih lanjut, lihat [Autentikasi multi-faktor](#) dalam Panduan Pengguna AWS IAM Identity Center dan [Menggunakan autentikasi multi-faktor \(MFA\) di AWS](#) dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari Anda. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar tugas lengkap yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya andalkan kredensial temporer, dan bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, sebaiknya rotasikan kunci akses. Untuk informasi selengkapnya, lihat [Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang](#) dalam Panduan Pengguna IAM.

[Grup IAM](#) adalah identitas yang menentukan kumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran tersebut dimaksudkan untuk dapat diambil oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, silakan lihat [Kapan harus membuat pengguna IAM \(bukan peran\)](#) dalam Panduan Pengguna IAM.

Peran IAM

[Peran IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan [beralih peran](#). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang metode untuk menggunakan peran, lihat [Menggunakan peran IAM](#) dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas gabungan, Anda dapat membuat peran dan menentukan izin untuk peran tersebut. Saat identitas terfederasi diautentikasi, identitas tersebut dikaitkan dengan peran dan diberikan izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) dalam Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda mengonfigurasi sekumpulan izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengaitkan izin yang ditetapkan ke peran dalam IAM. Untuk informasi tentang rangkaian izin, lihat [Rangkaian izin](#) dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (pengguna utama tepercaya) dengan akun berbeda untuk mengakses sumber daya yang ada di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Contoh, ketika Anda melakukan panggilan dalam layanan, umumnya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Suatu layanan mungkin melakukan hal tersebut menggunakan izin pengguna utama panggilan, menggunakan peran layanan, atau peran terkait layanan.
 - Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memulai tindakan lain

di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Meneruskan sesi akses](#).

- Peran IAM – Peran layanan adalah [peran IAM](#) yang diambil layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat mengambil peran untuk melakukan sebuah tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2](#) dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat [Kapan harus membuat peran IAM \(bukan pengguna\)](#) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat [Ikhtisar kebijakan JSON](#) dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat menjalankan peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk operasi. Sebagai contoh, anggap saja Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan pengguna dan peran, di sumber daya mana, dan dengan ketentuan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam Akun AWS. Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, lihat [Memilih antara kebijakan terkelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan tersebut, kebijakan ini menentukan jenis tindakan yang dapat dilakukan oleh pengguna utama tertentu di sumber daya tersebut dan apa ketentuannya. Anda harus [menentukan pengguna utama](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL sama dengan kebijakan berbasis sumber daya, meskipun tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, silakan lihat [Gambaran umum daftar kontrol akses \(ACL\)](#) di Panduan Developer Layanan Penyimpanan Ringkas Amazon.

Tipe kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda berdasarkan tipe kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan di mana Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan secara eksplisit terhadap salah satu kebijakan ini akan mengesampingkan izin tersebut. Untuk informasi selengkapnya tentang batasan izin, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) — SCP adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke sebagian atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat [Cara kerja SCP](#) dalam Panduan Pengguna AWS Organizations .
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda teruskan sebagai parameter saat Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit

di salah satu kebijakan ini akan membatalkan izin tersebut. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Jika beberapa jenis kebijakan diberlakukan untuk satu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat [Logika evaluasi kebijakan](#) di Panduan Pengguna IAM.

Bagaimana AWS Database Migration Service bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses AWS DMS, Anda harus memahami fitur IAM apa yang tersedia untuk digunakan. AWS DMS Untuk mendapatkan pandangan tingkat tinggi tentang bagaimana AWS DMS dan AWS layanan lain bekerja dengan IAM, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Topik

- [Kebijakan berbasis identitas AWS DMS](#)
- [AWS DMS Kebijakan berbasis sumber daya](#)
- [Otorisasi berdasarkan tanda AWS DMS](#)
- [Peran IAM untuk AWS DMS](#)
- [Manajemen identitas dan akses untuk DMS Fleet Advisor](#)

Kebijakan berbasis identitas AWS DMS

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, dan juga ketentuan di mana tindakan tersebut diperbolehkan atau ditolak. AWS DMS mendukung tindakan, sumber daya, dan kunci kondisi tertentu. Untuk mempelajari semua elemen yang Anda gunakan dalam kebijakan JSON, lihat [Referensi elemen kebijakan IAM JSON](#) dalam Panduan Pengguna IAM.

Tindakan

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam suatu kebijakan untuk memberikan izin melakukan operasi terkait.

Tindakan kebijakan AWS DMS menggunakan awalan berikut sebelum tindakan: `dms:`. Misalnya, untuk memberikan izin kepada seseorang untuk membuat tugas replikasi dengan operasi AWS DMS `CreateReplicationTask` API, Anda menyertakan `dms:CreateReplicationTask` tindakan tersebut dalam kebijakan mereka. Pernyataan kebijakan harus mencakup salah satu `Action` atau `NotAction` elemen. AWS DMS mendefinisikan serangkaian tindakannya sendiri yang menggambarkan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menentukan beberapa tindakan dalam satu pernyataan, pisahkan tindakan dengan koma seperti berikut:

```
"Action": [  
    "dms:action1",  
    "dms:action2"
```

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard (*). Misalnya, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut.

```
"Action": "dms:Describe*"
```

Untuk melihat daftar tindakan, lihat AWS DMS [Tindakan yang Ditentukan oleh AWS Database Migration Service](#) dalam Panduan Pengguna IAM.

Sumber daya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek atau beberapa objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat

melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk mengindikasikan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*" 
```

AWS DMS bekerja dengan sumber daya berikut:

- Sertifikat
- Titik akhir
- Langganan acara
- Instans replikasi
- Grup (keamanan) subnet replikasi
- Tugas replikasi

Sumber daya atau sumber daya yang AWS DMS dibutuhkan tergantung pada tindakan atau tindakan yang Anda panggil. Anda memerlukan kebijakan yang mengizinkan tindakan ini pada sumber daya terkait atau sumber daya yang ditentukan oleh ARN sumber daya.

Misalnya, sumber daya AWS DMS titik akhir memiliki ARN berikut:

```
arn:${Partition}:dms:${Region}:${Account}:endpoint/${InstanceId}
```

Untuk informasi selengkapnya tentang format ARN, lihat [Amazon Resource Names \(ARN\) dan ruang nama AWS layanan](#).

Misalnya, untuk menentukan instans titik akhir 1A2B3C4D5E6F7G8H9I0J1K2L3M untuk wilayah use-east-2 dalam pernyataan Anda, gunakan ARN berikut.

```
"Resource": "arn:aws:dms:us-east-2:987654321098:endpoint/1A2B3C4D5E6F7G8H9I0J1K2L3M" 
```

Untuk menentukan semua titik akhir milik akun tertentu, gunakan wildcard (*).

```
"Resource": "arn:aws:dms:us-east-2:987654321098:endpoint/*" 
```

Beberapa AWS DMS tindakan, seperti untuk membuat sumber daya, tidak dapat dilakukan pada sumber daya tertentu. Dalam kasus tersebut, Anda harus menggunakan wildcard (*).

```
"Resource": "*" 
```

Beberapa tindakan AWS DMS API melibatkan banyak sumber daya. Misalnya, `StartReplicationTask` memulai dan menghubungkan tugas replikasi ke dua sumber daya titik akhir basis data, yakni sumber dan target, sehingga pengguna IAM harus memiliki izin untuk membaca titik akhir sumber dan menulis ke titik akhir target. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARN dengan koma.

```
"Resource": [
  "resource1",
  "resource2" ]
```

Untuk informasi selengkapnya tentang mengontrol akses ke AWS DMS sumber daya menggunakan kebijakan, lihat [Menggunakan nama sumber daya untuk mengontrol akses](#). Untuk melihat daftar jenis sumber daya AWS DMS dan ARN mereka, lihat [Sumber Daya Ditetapkan oleh AWS Database Migration Service](#) di Panduan Pengguna IAM. Untuk mempelajari tindakan mana yang dapat menentukan ARN setiap sumber daya, lihat [Tindakan yang Ditentukan oleh Amazon AWS Database Migration Service](#).

Kunci syarat

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen `Condition` (atau blok `Condition`) memungkinkan Anda menentukan kondisi di mana suatu pernyataan akan diterapkan. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi kondisional yang menggunakan [operator kondisi](#), misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam satu pernyataan, atau beberapa kunci dalam satu elemen `Condition`, AWS akan mengevaluasinya dengan menggunakan operasi AND logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan OR operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika

izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, silakan lihat [Elemen kebijakan IAM: variabel dan tanda](#) di Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

AWS DMS mendefinisikan kumpulan kunci kondisinya sendiri dan juga mendukung penggunaan beberapa kunci kondisi global. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

AWS DMS mendefinisikan satu set tag standar yang dapat Anda gunakan dalam kunci kondisi dan juga memungkinkan Anda menentukan tag kustom Anda sendiri. Untuk informasi selengkapnya, lihat [Menggunakan tag untuk mengontrol akses](#).

Untuk melihat daftar kunci AWS DMS kondisi, lihat [Condition Keys untuk AWS Database Migration Service](#) di Panduan Pengguna IAM. Untuk mempelajari tindakan dan sumber daya di mana Anda dapat gunakan kunci kondisi, lihat [Tindakan yang Ditentukan oleh AWS Database Migration Service](#) dan [Sumber Daya yang Ditentukan oleh AWS Database Migration Service](#).

Contoh-contoh

Untuk melihat contoh kebijakan AWS DMS berbasis identitas, lihat [AWS Database Migration Service contoh kebijakan berbasis identitas](#)

AWS DMS Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada AWS DMS sumber daya tertentu dan dalam kondisi apa. AWS DMS mendukung kebijakan izin berbasis sumber daya untuk kunci AWS KMS enkripsi yang Anda buat untuk mengenkripsi data yang dimigrasi ke titik akhir target yang didukung. Titik akhir target yang didukung termasuk Amazon Redshift dan Amazon S3. Dengan menggunakan kebijakan berbasis sumber daya, Anda dapat memberikan izin untuk menggunakan kunci enkripsi ini ke akun lain untuk setiap titik akhir target.

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau entitas IAM di akun lain sebagai [prinsipal dalam kebijakan berbasis sumber daya](#). Menambahkan pengguna utama lintas akun ke kebijakan berbasis sumber daya bagian dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berada di AWS akun yang berbeda, Anda juga harus memberikan izin entitas utama untuk mengakses sumber daya. Berikan izin dengan melampirkan kebijakan

berbasis identitas ke entitas tersebut. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat [Perbedaan peran IAM dengan kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

AWS DMS Layanan ini hanya mendukung satu jenis kebijakan berbasis sumber daya yang disebut kebijakan kunci, yang dilampirkan ke kunci enkripsi. AWS KMS Kebijakan ini menentukan entitas prinsipal mana (akun, pengguna, peran, dan pengguna gabungan) yang dapat mengenkripsi data yang dimigrasikan di titik akhir target yang didukung.

Untuk mempelajari cara melampirkan kebijakan berbasis sumber daya ke kunci enkripsi yang Anda buat untuk titik akhir target yang didukung, lihat [Membuat dan menggunakan kunci AWS KMS untuk mengenkripsi data target Amazon Redshift](#) dan [Membuat kunci AWS KMS untuk mengenkripsi objek target Amazon S3](#).

Contoh-contoh

Untuk contoh kebijakan AWS DMS berbasis sumber daya, lihat. [Contoh kebijakan berbasis sumber daya untuk AWS KMS](#)

Otorisasi berdasarkan tanda AWS DMS

Anda dapat melampirkan tag ke AWS DMS sumber daya atau meneruskan tag dalam permintaan AWS DMS. Untuk mengontrol akses berdasarkan tag, Anda memberikan informasi tag dalam [elemen kondisi](#) kebijakan menggunakan `dms:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau kunci `aws:TagKeys` kondisi. AWS DMS mendefinisikan satu set tag standar yang dapat Anda gunakan dalam kunci kondisi dan juga memungkinkan Anda untuk menentukan tag kustom Anda sendiri. Untuk informasi selengkapnya, lihat [Menggunakan tag untuk mengontrol akses](#).

Untuk melihat contoh kebijakan berbasis identitas yang membatasi akses ke sumber daya berdasarkan tag, lihat [Mengakses sumber daya AWS DMS berdasarkan tag](#).

Peran IAM untuk AWS DMS

[Peran IAM](#) adalah entitas dalam AWS akun Anda yang memiliki izin tertentu.

Menggunakan kredensial sementara dengan AWS DMS

Anda dapat menggunakan kredensial sementara untuk masuk dengan federasi, memiliki IAM role, atau menjalankan peran lintas-akun. Anda mendapatkan kredensial keamanan sementara dengan memanggil operasi AWS STS API seperti [AssumeRole](#) atau [GetFederationToken](#)

AWS DMS mendukung menggunakan kredensial sementara.

Peran terkait layanan

[Peran terkait AWS layanan](#) memungkinkan layanan mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran terkait layanan muncul di akun IAM Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran AWS DMS terkait layanan, lihat [Menggunakan peran terkait layanan](#)

Peran layanan

Fitur ini memungkinkan layanan untuk menerima [peran layanan](#) atas nama Anda. Peran ini mengizinkan layanan untuk mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran layanan muncul di akun IAM Anda dan dimiliki oleh akun tersebut. Ini berarti administrator IAM dapat mengubah izin untuk peran ini. Namun, melakukan hal itu dapat merusak fungsionalitas layanan.

AWS DMS mendukung dua jenis peran layanan yang harus Anda buat untuk menggunakan sumber atau titik akhir target tertentu:

- Peran dengan izin untuk memungkinkan akses AWS DMS ke sumber dan titik akhir target berikut (atau sumber dayanya):
 - Amazon DynamoDB sebagai target — Untuk informasi lebih lanjut lihat [Prasyarat untuk menggunakan DynamoDB sebagai target untuk AWS Database Migration Service](#).
 - OpenSearch sebagai target — Untuk informasi lebih lanjut lihat [Prasyarat untuk menggunakan AmazonOpenSearchLayanan sebagai targetAWS Database Migration Service](#).
 - Amazon Kinesis sebagai target – Untuk informasi lebih lanjut lihat [Prasyarat untuk menggunakan aliran data Kinesis sebagai target AWS Database Migration Service](#).
 - Amazon Redshift sebagai target – Anda perlu membuat peran tertentu hanya untuk menciptakan kunci enkripsi KMS kustom untuk mengenkripsi data target atau untuk menentukan bucket S3 kustom untuk memegang penyimpanan tugas menengah. Untuk informasi lebih lanjut, lihat [Membuat dan menggunakan kunci AWS KMS untuk mengenkripsi data target Amazon Redshift](#) atau [Pengaturan bucket Amazon S3](#).
 - Amazon S3 sebagai sumber atau sebagai target – Untuk informasi lebih lanjut, lihat [Prasyarat saat menggunakan Amazon S3 sebagai sumber untuk AWS DMS](#) atau [Prasyarat penggunaan Amazon S3 sebagai target](#).

Misalnya, untuk membaca data dari titik akhir sumber S3 atau untuk mendorong data ke titik akhir target S3, Anda harus membuat peran layanan sebagai prasyarat untuk mengakses S3 untuk masing-masing operasi titik akhir ini.

- Peran dengan izin yang diperlukan untuk menggunakan API AWS CLI dan AWS DMS — Dua peran IAM yang perlu Anda buat adalah `aws-logs-vpc-role` dan `aws-logs-cloudwatch-logs-role`. Jika Anda menggunakan Amazon Redshift sebagai database target, Anda juga harus membuat dan menambahkan peran IAM `aws-logs-access-for-endpoint` ke akun Anda. AWS Untuk informasi selengkapnya, lihat [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#).

Memilih peran IAM di AWS DMS

Jika Anda menggunakan AWS CLI atau AWS DMS API untuk migrasi database Anda, Anda harus menambahkan peran IAM tertentu ke AWS akun Anda sebelum Anda dapat menggunakan fitur DMS. AWS Dua dari peran tersebut adalah `aws-logs-vpc-role` dan `aws-logs-cloudwatch-logs-role`. Jika Anda menggunakan Amazon Redshift sebagai database target, Anda juga harus menambahkan peran IAM `aws-logs-access-for-endpoint` ke akun Anda. AWS Untuk informasi selengkapnya, lihat [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#).

Manajemen identitas dan akses untuk DMS Fleet Advisor

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, dan juga kondisi di mana tindakan diizinkan atau ditolak. DMS Fleet Advisor mendukung tindakan, sumber daya, dan kunci kondisi tertentu. Untuk mempelajari semua elemen yang Anda gunakan dalam kebijakan JSON, lihat [Referensi elemen kebijakan IAM JSON](#) dalam Panduan Pengguna IAM.

DMS Fleet Advisor menggunakan peran IAM untuk mengakses Amazon Simple Storage Service. [Peran IAM](#) adalah entitas dalam AWS akun Anda yang memiliki izin tertentu. Untuk informasi selengkapnya, lihat [Buat sumber daya IAM](#).

AWS Database Migration Service contoh kebijakan berbasis identitas

Secara default, pengguna dan peran IAM tidak memiliki izin untuk membuat atau memodifikasi sumber daya AWS DMS. Mereka juga tidak dapat melakukan tugas menggunakan AWS Management Console, AWS CLI, atau AWS API. Administrator IAM harus membuat kebijakan IAM yang memberikan izin kepada pengguna dan peran untuk melakukan operasi API tertentu pada sumber daya yang diperlukan. Administrator kemudian harus melampirkan kebijakan tersebut ke pengguna IAM atau grup yang memerlukan izin tersebut.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan di tab JSON](#) dalam Panduan Pengguna IAM.

Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan konsol AWS DMS](#)
- [Izinkan pengguna melihat izin mereka sendiri](#)
- [Mengakses satu bucket Amazon S3](#)
- [Mengakses sumber daya AWS DMS berdasarkan tag](#)

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus AWS DMS sumber daya di akun Anda. Tindakan ini dikenai biaya untuk Akun AWS Anda. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat [kebijakan yang dikelola AWS](#) atau [kebijakan yang dikelola AWS untuk fungsi pekerjaan](#) di Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukan ini dengan menentukan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk menerapkan izin, lihat [Kebijakan dan izin di IAM](#) di Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Misalnya, Anda dapat menulis syarat kebijakan untuk menentukan bahwa semua pengajuan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS

CloudFormation. Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Syarat](#) di Panduan Pengguna IAM.

- Menggunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda guna memastikan izin yang aman dan berfungsi – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat [validasi kebijakan Analizer Akses IAM](#) di Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk mewajibkan MFA saat operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat [Mengonfigurasi akses API yang dilindungi MFA](#) di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan di IAM](#) di Panduan Pengguna IAM.

Menggunakan konsol AWS DMS

Kebijakan berikut memberi Anda akses ke AWS DMS, termasuk konsol AWS DMS, dan juga menentukan izin untuk tindakan tertentu yang diperlukan dari layanan Amazon lainnya seperti 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"
    }
  ]
}
```

```
}
```

Perincian izin ini dapat membantu Anda lebih memahami mengapa masing-masing izin diperlukan untuk menggunakan konsol tersebut.

Bagian berikut ini diperlukan untuk mengizinkan pengguna untuk membuat daftar kunci AWS KMS yang tersedia dan alias untuk ditampilkan di konsol. Entri ini tidak diperlukan jika Anda mengetahui Amazon Resource Name (ARN) untuk kunci KMS dan Anda hanya menggunakan AWS Command Line Interface (AWS CLI).

```
{
    "Effect": "Allow",
    "Action": [
        "kms:ListAliases",
        "kms:DescribeKey"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}
```

Bagian berikut ini diperlukan untuk jenis titik akhir tertentu yang memerlukan ARN peran untuk diteruskan dengan titik akhir. Selain itu, jika AWS DMS peran yang diperlukan tidak dibuat sebelumnya, AWS DMS konsol memiliki kemampuan untuk membuat peran. Jika semua peran dikonfigurasi sebelumnya, yang diperlukan adalah `iam:GetRole` dan `iam:PassRole`. Untuk informasi lebih lanjut tentang peran, lihat [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#).

```
{
    "Effect": "Allow",
    "Action": [
        "iam:GetRole",
        "iam:PassRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}
```

Bagian berikut diperlukan karena AWS DMS perlu membuat instans Amazon EC2 dan mengonfigurasi jaringan untuk instance replikasi yang dibuat. Sumber daya ini ada di akun pelanggan, sehingga kemampuan untuk melakukan tindakan ini atas nama pelanggan diperlukan.

```
{
  "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"
}
```

Bagian berikut diperlukan untuk mengizinkan pengguna untuk dapat melihat metrik instans replikasi.

```
{
  "Effect": "Allow",
  "Action": [
    "cloudwatch:Get*",
    "cloudwatch:List*"
  ],
  "Resource": "arn:aws:service:region:account:resourcetype/id"
}
```

Bagian ini diperlukan untuk mengizinkan pengguna untuk melihat log replikasi.

```
{
  "Effect": "Allow",
  "Action": [
    "logs:DescribeLogGroups",
    "logs:DescribeLogStreams",
    "logs:FilterLogEvents",
    "logs:GetLogEvents"
  ],
  "Resource": "arn:aws:service:region:account:resourcetype/id"
}
```

Konsol AWS DMS membuat beberapa peran yang secara otomatis dilampirkan ke AWS akun Anda saat Anda menggunakan konsol AWS DMS. Jika Anda menggunakan AWS Command Line Interface (AWS CLI) atau AWS DMS API untuk migrasi, Anda perlu menambahkan peran ini ke akun Anda. Untuk informasi lebih lanjut tentang menambahkan peran ini, lihat [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#).

Untuk informasi selengkapnya tentang persyaratan penggunaan kebijakan ini untuk mengakses AWS DMS, lihat [Izin IAM diperlukan untuk menggunakan AWS DMS](#).

Izinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan para pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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": "*"
    }
  ]
}
```

Mengakses satu bucket Amazon S3

AWS DMS menggunakan bucket Amazon S3 sebagai penyimpanan perantara untuk migrasi database. Biasanya, AWS DMS mengelola bucket S3 default untuk tujuan ini. Namun, dalam kasus tertentu, terutama ketika Anda menggunakan AWS CLI atau AWS DMS API, AWS DMS memungkinkan Anda untuk menentukan bucket S3 Anda sendiri sebagai gantinya. Misalnya, Anda dapat menentukan bucket S3 Anda sendiri untuk memigrasi data ke titik akhir target Amazon Redshift. Dalam hal ini, Anda perlu membuat peran dengan izin berdasarkan kebijakan AWS - `managedAmazonDMSRedshiftS3Role`.

Contoh berikut menunjukkan sebuah versi kebijakan `AmazonDMSRedshiftS3Role`. Hal ini memungkinkan AWS DMS untuk memberikan pengguna IAM di AWS akun Anda akses ke salah satu bucket Amazon S3 Anda. Hal ini juga memungkinkan pengguna untuk menambahkan, memperbaiki, dan menghapus objek.

Selain memberikan izin `s3:PutObject`, `s3:GetObject`, dan `s3:DeleteObject` bagi pengguna, kebijakan tersebut juga memberikan izin `s3:ListAllMyBuckets`, `s3:GetBucketLocation`, dan `s3:ListBucket`. Izin-izin tersebut adalah izin tambahan yang diperlukan oleh konsol tersebut. Izin lain memungkinkan AWS DMS mengelola siklus hidup bucket. Selain itu, tindakan `s3:GetObjectAcl` diperlukan untuk dapat menyalin objek.

```
{
  "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-*"
}
]
}

```

Untuk informasi lebih lanjut tentang cara membuat peran berdasarkan kebijakan ini, lihat [Pengaturan bucket Amazon S3](#).

Mengakses sumber daya AWS DMS berdasarkan tag

Anda dapat menggunakan ketentuan dalam kebijakan berbasis identitas Anda untuk mengontrol akses ke sumber daya AWS DMS berdasarkan tag. Contoh ini menunjukkan cara membuat kebijakan yang memungkinkan akses ke semua titik akhir AWS DMS. Namun, izin diberikan hanya jika tag basis data titik akhir `Owner` memiliki nilai nama pengguna dari pengguna tersebut.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "dms:*",
      "Resource": "arn:aws:dms:*:*:endpoint/*",
      "Condition": {
        "StringEquals": {"dms:endpoint-tag/Owner": "${aws:username}"}
      }
    }
  ]
}

```

Anda dapat melampirkan kebijakan ini ke pengguna IAM di akun Anda. Jika pengguna bernama `richard-roe` mencoba mengakses AWS DMS titik akhir, database endpoint harus diberi tag `Owner=richard-roe` atau `owner=richard-roe`. Jika tidak, akses pengguna ini ditolak. Kunci tag

kondisi Owner cocok dengan Owner dan owner karena nama kunci kondisi tidak terpengaruh huruf besar/kecil. Untuk informasi selengkapnya, lihat [Elemen kebijakan IAM JSON: Syarat](#) dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis sumber daya untuk AWS KMS

AWS DMS memungkinkan Anda membuat kunci AWS KMS enkripsi khusus untuk mengenkripsi data titik akhir target yang didukung. Untuk mempelajari cara membuat dan melampirkan kebijakan kunci ke kunci enkripsi yang Anda buat untuk mengenkripsi data target yang didukung, lihat [Membuat dan menggunakan kunci AWS KMS untuk mengenkripsi data target Amazon Redshift](#) dan [Membuat kunci AWS KMS untuk mengenkripsi objek target Amazon S3](#).

Topik

- [Kebijakan untuk kunci AWS KMS enkripsi khusus untuk mengenkripsi data target Amazon Redshift](#)
- [Kebijakan untuk kunci AWS KMS enkripsi khusus untuk mengenkripsi data target Amazon S3](#)

Kebijakan untuk kunci AWS KMS enkripsi khusus untuk mengenkripsi data target Amazon Redshift

Contoh berikut menunjukkan JSON untuk kebijakan kunci yang dibuat untuk kunci enkripsi AWS KMS yang Anda buat untuk mengenkripsi data target 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
    }
  }
}
]
}

```

Di sini, Anda dapat melihat di mana kebijakan kunci mereferensikan peran untuk mengakses data titik akhir target Amazon Redshift yang Anda buat sebelum membuat kunci. Dalam contoh, yaitu `DMS-Redshift-endpoint-access-role`. Anda juga dapat melihat berbagai tindakan kunci yang diizinkan untuk prinsipal yang berbeda (pengguna dan peran). Misalnya, setiap pengguna dengan `DMS-Redshift-endpoint-access-role` dapat mengenkripsi, mendekripsi, dan mengenkripsi ulang data target. Pengguna tersebut juga dapat menghasilkan kunci data untuk ekspor untuk mengenkripsi data di luar. AWS KMS Mereka juga dapat mengembalikan informasi terperinci tentang AWS KMS kunci, seperti kunci yang baru saja Anda buat. Selain itu, pengguna semacam itu dapat mengelola lampiran sumber daya AWS , seperti titik akhir target.

Kebijakan untuk kunci AWS KMS enkripsi khusus untuk mengenkripsi data target Amazon S3

Contoh berikut menunjukkan JSON untuk kebijakan kunci yang dibuat untuk kunci enkripsi AWS KMS yang Anda buat untuk mengenkripsi data target 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
    }
  }
}
]

```

Di sini, Anda dapat melihat di mana kebijakan kunci mereferensikan peran untuk mengakses data titik akhir target Amazon S3 yang Anda buat sebelum membuat kunci. Dalam contoh, yaitu `DMS-S3-endpoint-access-role`. Anda juga dapat melihat berbagai tindakan kunci yang diizinkan untuk prinsipal yang berbeda (pengguna dan peran). Misalnya, setiap pengguna dengan `DMS-S3-endpoint-access-role` dapat mengenkripsi, mendekripsi, dan mengenkripsi ulang data target. Pengguna tersebut juga dapat menghasilkan kunci data untuk ekspor untuk mengenkripsi data di luar. AWS KMS Mereka juga dapat mengembalikan informasi terperinci tentang AWS KMS kunci, seperti kunci yang baru saja Anda buat. Selain itu, pengguna semacam itu dapat mengelola lampiran sumber daya AWS, seperti titik akhir target.

Menggunakan rahasia untuk mengakses titik akhir AWS Database Migration Service

Untuk AWS DMS, rahasia adalah kunci terenkripsi yang dapat Anda gunakan untuk mewakili sekumpulan kredensi pengguna untuk mengautentikasi, melalui otentikasi rahasia, koneksi database untuk sumber yang didukung atau titik akhir target. AWS DMS Untuk endpoint Oracle yang juga menggunakan Oracle Automatic Storage Management (ASM), AWS DMS memerlukan rahasia tambahan yang mewakili kredensi pengguna untuk mengakses Oracle ASM.

Anda dapat membuat rahasia atau rahasia yang AWS DMS memerlukan otentikasi rahasia menggunakan AWS Secrets Manager, layanan untuk membuat, menyimpan, dan mengambil kredensial dengan aman untuk mengakses aplikasi, layanan, dan sumber daya TI di cloud dan di lokasi. Ini termasuk dukungan untuk rotasi berkala otomatis dari nilai rahasia terenkripsi tanpa intervensi Anda, sehingga memberikan tingkat keamanan tambahan untuk kredensial Anda. Mengaktifkan rotasi nilai rahasia AWS Secrets Manager juga memastikan bahwa rotasi nilai rahasia ini terjadi tanpa efek apa pun pada migrasi database apa pun yang bergantung pada rahasia. Untuk mengautentikasi koneksi basis data titik akhir secara rahasia, buat rahasia yang identitas atau ARN-nya Anda tetapkan untuk `SecretsManagerSecretId`, yang Anda sertakan dalam pengaturan titik akhir Anda. Untuk mengautentikasi Oracle ASM sebagai bagian dari titik akhir Oracle secara rahasia, buat rahasia yang identitas atau ARN-nya Anda tetapkan untuk `SecretsManagerOracleAsmSecretId`, yang juga Anda sertakan dalam pengaturan titik akhir Anda.

Note

Anda tidak dapat menggunakan kredensial master yang dikelola oleh Amazon RDS Aurora. Kredensi ini tidak termasuk informasi host atau port, yang AWS DMS perlu membuat koneksi. Sebagai gantinya, buat pengguna baru dan rahasia. Untuk informasi tentang membuat pengguna dan rahasia, lihat [Menggunakan AWS Management Console untuk membuat peran akses rahasia dan rahasia](#) berikut.

Untuk informasi selengkapnya AWS Secrets Manager, lihat [Apa itu AWS Secrets Manager?](#) dalam AWS Secrets Manager User Guide.

AWS DMS mendukung otentikasi rahasia untuk database on-premise atau AWS -managed berikut pada sumber yang didukung dan titik akhir target:

- Amazon DocumentDB
- IBM Db2 LUW
- Microsoft SQL Server
- MongoDB
- MySQL
- Oracle
- PostgreSQL
- Amazon Redshift
- SAP ASE

Untuk koneksi ke salah satu basis data ini, Anda memiliki pilihan untuk memasukkan salah satu dari rangkaian nilai berikut, tetapi tidak keduanya, sebagai bagian dari pengaturan titik akhir Anda:

- Nilai teks yang jelas untuk mengautentikasi koneksi basis data menggunakan pengaturan `UserName`, `Password`, `ServerName`, dan `Port`. Untuk titik akhir Oracle yang juga menggunakan Oracle ASM, sertakan nilai teks yang jelas tambahan untuk mengautentikasi ASM menggunakan pengaturan `AsmUserName`, `AsmPassword`, dan `AsmServerName`.
- Autentikasi rahasia menggunakan nilai untuk pengaturan `SecretsManagerSecretId` dan `SecretsManagerAccessRoleArn`. Untuk titik akhir Oracle yang menggunakan Oracle ASM, sertakan nilai tambahan untuk pengaturan `SecretsManagerOracleAsmSecretId` dan `SecretsManagerOracleAsmAccessRoleArn`. Nilai rahasia untuk pengaturan ini dapat mencakup berikut ini untuk:
 - `SecretsManagerSecretId` – Amazon Resource Name (ARN) lengkap, ARN parsial, atau nama yang dikenal dari rahasia yang telah Anda buat untuk akses basis data titik akhir di AWS Secrets Manager.
 - `SecretsManagerAccessRoleArn`— ARN dari peran akses rahasia yang telah Anda buat di IAM untuk menyediakan AWS DMS akses ke `SecretsManagerSecretId` rahasia ini atas nama Anda.
 - `SecretsManagerOracleAsmSecretId` – Amazon Resource Name (ARN) lengkap, ARN parsial, atau nama yang dikenal dari rahasia yang telah Anda buat untuk akses Oracle ASM di AWS Secrets Manager.
 - `SecretsManagerOracleAsmAccessRoleArn` – ARN dari peran akses rahasia yang telah Anda buat di IAM untuk memberikan akses AWS DMS ke rahasia `SecretsManagerOracleAsmSecretId` atas nama Anda.

Note

Anda juga dapat menggunakan peran akses rahasia tunggal untuk menyediakan AWS DMS akses ke `SecretsManagerSecretId` rahasia dan `SecretsManagerOracleAsmSecretId` rahasia. Jika Anda membuat peran akses rahasia tunggal ini untuk kedua rahasia, pastikan bahwa Anda menetapkan ARN yang sama untuk peran akses ini untuk `SecretsManagerAccessRoleArn` dan `SecretsManagerOracleAsmAccessRoleArn`. Misalnya, jika peran akses rahasia Anda untuk kedua rahasia memiliki ARN yang ditugaskan untuk variabel, `ARN2xsecrets`, Anda dapat mengatur pengaturan ARN ini sebagai berikut:

```
SecretsManagerAccessRoleArn = ARN2xsecrets;  
SecretsManagerOracleAsmAccessRoleArn = ARN2xsecrets;
```

Untuk informasi lebih lanjut tentang nilai-nilai tersebut, lihat [Menggunakan AWS Management Console untuk membuat peran akses rahasia dan rahasia](#).

Setelah Anda telah membuat dan menentukan pengaturan rahasia dan titik akhir akses-peran rahasia yang diperlukan untuk titik akhir Anda, perbarui izin pada akun pengguna yang akan menjalankan permintaan `CreateEndpoint` atau `ModifyEndpoint` API dengan detail rahasia ini. Pastikan bahwa izin akun ini mencakup `IAM:GetRole` izin pada peran akses rahasia dan `SecretsManager:DescribeSecret` izin pada rahasia. AWS DMS memerlukan izin ini untuk memvalidasi peran akses dan rahasianya.

Untuk memberikan dan memverifikasi izin pengguna yang diperlukan

1. Masuk ke AWS Management Console dan buka AWS Identity and Access Management konsol di <https://console.aws.amazon.com/iam/>.
2. Pilih Pengguna, lalu pilih ID Pengguna yang digunakan untuk membuat panggilan `CreateEndpoint` dan `ModifyEndpoint` API.
3. Dari tab Izin, pilih `{}` JSON.
4. Pastikan pengguna memiliki izin yang ditampilkan berikut ini.

```
{  
  "Statement": [{
```

```
"Effect": "Allow",
"Action": [
  "iam:GetRole",
  "iam:PassRole"
],
"Resource": "SECRET_ACCESS_ROLE_ARN"
},
{
  "Effect": "Allow",
  "Action": "secretsmanager:DescribeSecret",
  "Resource": "SECRET_ARN"
}
]
```

5. Jika pengguna tidak memiliki izin tersebut, tambahkan izin.
6. Jika Anda menggunakan IAM role untuk melakukan panggilan DMS API, ulangi langkah-langkah di atas untuk masing-masing peran.
7. Buka terminal dan gunakan AWS CLI untuk memvalidasi bahwa izin diberikan dengan benar dengan mengasumsikan Peran atau Pengguna yang digunakan di atas.
 - a. Validasi izin pengguna pada SecretAccessRole menggunakan perintah `IAMget-role`.

```
aws iam get-role --role-name ROLE_NAME
```

Ganti *ROLE_NAME* dengan nama `SecretsManagerAccessRole`.

Jika perintah mengembalikan pesan kesalahan, pastikan izin diberikan dengan benar.

- b. Memvalidasi izin pengguna pada rahasia menggunakan perintah `Secrets Manager describe-secret`.

```
aws secretsmanager describe-secret --secret-id SECRET_NAME OR SECRET_ARN --
region=REGION_NAME
```

Pengguna bisa menjadi nama yang dikenal, ARN parsial atau ARN lengkap. Untuk informasi lebih lanjut, lihat [uraikan-rahasia](#).

Jika perintah mengembalikan pesan kesalahan, pastikan izin diberikan dengan benar.

Menggunakan AWS Management Console untuk membuat peran akses rahasia dan rahasia

Anda dapat menggunakan AWS Management Console untuk membuat rahasia untuk otentikasi titik akhir dan untuk membuat kebijakan dan peran AWS DMS untuk memungkinkan mengakses rahasia atas nama Anda.

Untuk membuat rahasia menggunakan AWS Management Console yang AWS DMS dapat digunakan untuk mengautentikasi database untuk sumber dan koneksi titik akhir target

1. Masuk ke AWS Management Console dan buka AWS Secrets Manager konsol di <https://console.aws.amazon.com/secretsmanager/>.
2. Pilih Simpan rahasia baru.
3. Di bawah Pilih tipe rahasia pada halaman Simpan rahasia baru, pilih Jenis rahasia lainnya, lalu pilih Plaintext.

Note

Ini adalah satu-satunya tempat yang Anda butuhkan untuk memasukkan kredensial teks yang jelas untuk terhubung ke basis data titik akhir Anda mulai saat ini hingga seterusnya.

4. Di bidang Plaintext:
 - Untuk rahasia yang identitasnya Anda tetapkan pada `SecretsManagerSecretId`, masukkan struktur JSON berikut ini.

```
{
  "username": db_username,
  "password": db_user_password,
  "port": db_port_number,
  "host": db_server_name
}
```

Note

Ini adalah daftar minimum anggota JSON yang diperlukan untuk mengautentikasi basis data titik akhir. Anda dapat menambahkan pengaturan titik akhir JSON tambahan sebagai anggota JSON dalam huruf kecil yang Anda inginkan. Namun, AWS DMS mengabaikan setiap anggota JSON tambahan untuk autentikasi titik akhir.

Di sini, *db_username* adalah nama pengguna yang mengakses basis data , *db_user_password* adalah kata sandi dari pengguna basis data, *db_port_number* adalah nomor port untuk mengakses database, dan *db_server_name* adalah nama server basis data (alamat) di web, seperti dalam contoh berikut.

```
{
  "username": "admin",
  "password": "some_password",
  "port": "8190",
  "host": "oracle101.abcdefghij.us-east-1.rds.amazonaws.com"
}
```

- Untuk rahasia yang identitasnya Anda tetapkan pada `SecretsManagerOracleAsmSecretId`, masukkan struktur JSON berikut ini.

```
{
  "asm_user": asm_username,
  "asm_password": asm_user_password,
  "asm_server": asm_server_name
}
```

Note

Ini adalah daftar minimum anggota JSON yang diperlukan untuk mengautentikasi Oracle ASM untuk titik akhir Oracle. Daftar ini juga merupakan daftar lengkap yang dapat Anda tentukan berdasarkan pengaturan titik akhir Oracle ASM yang tersedia.

Di sini, *asm_username* adalah nama pengguna yang mengakses Oracle ASM, *asm_user_password* adalah kata sandi pengguna Oracle ASM, dan *asm_server_name* adalah nama server (alamat) Oracle ASM di web, termasuk port, seperti dalam contoh berikut.

```
{
  "asm_user": "oracle_asm_user",
  "asm_password": "oracle_asm_password",
  "asm_server": "oracle101.abcdefghij.us-east-1.rds.amazonaws.com:8190/+ASM"
}
```

5. Pilih kunci AWS KMS enkripsi untuk mengenkripsi rahasia. Anda dapat menerima kunci enkripsi default yang dibuat untuk layanan Anda dengan AWS Secrets Manager atau memilih AWS KMS kunci yang Anda buat.
6. Tentukan nama untuk mereferensikan rahasia ini dan deskripsi opsional. Ini adalah nama yang dikenal yang Anda gunakan sebagai nilai untuk `SecretsManagerSecretId` atau `SecretsManagerOracleAsmSecretId`.
7. Jika Anda ingin mengaktifkan rotasi otomatis dalam rahasia, Anda perlu memilih atau membuat AWS Lambda fungsi dengan izin untuk memutar kredensial untuk rahasia seperti yang dijelaskan. Namun, sebelum mengatur rotasi otomatis untuk menggunakan fungsi Lambda Anda, pastikan bahwa pengaturan konfigurasi untuk fungsi tersebut menambahkan empat karakter berikut ke nilai variabel lingkungan `EXCLUDE_CHARACTERS`.

```
;.:+{}
```

AWS DMS tidak mengizinkan karakter ini dalam kata sandi yang digunakan untuk kredensial titik akhir. Mengonfigurasi fungsi Lambda Anda untuk mengecualikan karakter tersebut mencegah AWS Secrets Manager dari menghasilkan karakter tersebut sebagai bagian dari nilai kata sandi yang dirotasi. Setelah Anda mengatur rotasi otomatis untuk menggunakan fungsi Lambda Anda, AWS Secrets Manager segera putar rahasia untuk memvalidasi konfigurasi rahasia Anda.

Note

Tergantung pada konfigurasi mesin basis data Anda, basis data Anda mungkin tidak mengambil kredensial yang dirotasi. Dalam hal ini, Anda perlu memulai ulang tugas secara manual untuk menyegarkan kredensial.

8. Tinjau dan simpan rahasia Anda AWS Secrets Manager. Anda kemudian dapat mencari setiap rahasia dengan nama ramahnya AWS Secrets Manager, lalu mengambil ARN rahasia sebagai nilai `SecretsManagerSecretId` untuk `SecretsManagerOracleAsmSecretId` atau sesuai untuk mengautentikasi akses ke koneksi database endpoint Anda dan Oracle ASM (jika digunakan).

Untuk membuat kebijakan akses rahasia dan peran untuk mengatur

SecretsManagerAccessRoleArn atau **SecretsManagerOracleAsmAccessRoleArn**, yang memungkinkan akses AWS DMS ke AWS Secrets Manager untuk mengakses rahasia Anda yang sesuai

1. Masuk ke AWS Management Console dan buka konsol AWS Identity and Access Management (IAM) di <https://console.aws.amazon.com/iam/>.
2. Pilih Kebijakan, lalu pilih Buat kebijakan.
3. Pilih JSON dan masukkan kebijakan berikut untuk mengaktifkan akses ke dan dekripsi rahasia Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": secret_arn,
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:DescribeKey"
      ],
      "Resource": kms_key_arn,
    }
  ]
}
```

Di sini, *secret_arn* adalah ARN rahasia Anda, yang bisa Anda dapatkan dari `SecretsManagerSecretId` atau `SecretsManagerOracleAsmSecretId` yang sesuai, dan *kms_key_arn* adalah ARN kunci AWS KMS yang Anda gunakan untuk mengenkripsi rahasia Anda, seperti dalam contoh berikut.

```
{
  "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

Jika Anda menggunakan kunci enkripsi default yang dibuat oleh AWS Secrets Manager, Anda tidak perlu menentukan AWS KMS izin untuk *kms_key_arn*.

Jika Anda ingin kebijakan Anda untuk menyediakan akses ke kedua rahasia, cukup tentukan objek sumber daya JSON tambahan untuk *secret_arn* lainnya.

Jika rahasia Anda ada di akun yang berbeda, maka `SecretsManagerAccessRoleArn` peran tersebut memerlukan kebijakan tambahan untuk memverifikasi rahasia lintas akun. Untuk kasus penggunaan seperti itu, tambahkan tindakan `secretsmanager:DescribeSecret` ke kebijakan. Untuk detail selengkapnya tentang menyiapkan rahasia lintas akun, lihat Rahasia [Izin untuk AWS Secrets Manager untuk pengguna di akun lain](#).

4. Tinjau dan buat kebijakan dengan nama yang dikenal dan deskripsi opsional.
5. Pilih Peran, lalu pilih Buat peran.
6. Pilih layanan AWS sebagai jenis entitas terpercaya.
7. Pilih DMS dari daftar layanan sebagai layanan terpercaya, lalu pilih Berikutnya: Izin.

8. Cari dan lampirkan kebijakan yang Anda buat di langkah 4, lalu lanjutkan dengan menambahkan tag dan tinjau peran Anda. Pada titik ini, edit hubungan kepercayaan agar peran tersebut menggunakan prinsipal layanan AWS DMS regional Anda sebagai entitas tepercaya. Prinsipal ini memiliki format berikut.

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

Di sini, *region-name* adalah nama wilayah Anda, seperti `us-east-1`. Dengan demikian, kepala layanan AWS DMS regional untuk wilayah ini mengikuti.

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

9. Setelah mengedit entitas tepercaya untuk peran tersebut, buat peran dengan nama yang dikenal dan deskripsi opsional. Anda sekarang dapat mencari peran baru Anda dengan nama yang dikenal di IAM, kemudian mengambil ARN peran sebagai nilai `SecretsManagerAccessRoleArn` atau `SecretsManagerOracleAsmAccessRoleArn` untuk mengautentikasi koneksi basis data titik akhir Anda.

Untuk menggunakan secrets manager dengan instans replikasi di subnet privat

1. Buat VPC endpoint secret manager dan perhatikan DNS untuk titik akhir. Untuk informasi lebih lanjut tentang cara membuat VPC endpoint secret manager, lihat [Menghubungkan ke Secrets Manager melalui VPC endpoint](#) di Panduan Pengguna Secrets Manager AWS .
2. Lampirkan grup keamanan instans replikasi ke VPC endpoint secret manager.
3. Untuk aturan keluar grup keamanan instans replikasi, izinkan semua lalu lintas untuk tujuan `0.0.0.0/0`.
4. Atur atribut koneksi tambahan titik akhir, `secretsManagerEndpointOverride=secretsManager endpoint DNS` untuk memberikan DNS VPC endpoint secret manager, seperti yang ditunjukkan dalam contoh berikut.

```
secretsManagerEndpointOverride=vpce-1234a5678b9012c-12345678.secretsmanager.eu-west-1.vpce.amazonaws.com
```

Menggunakan peran terkait layanan untuk AWS DMS

AWS Database Migration Service menggunakan AWS Identity and Access Management peran yang terkait dengan layanan [\(IAM\)](#). Peran yang terkait dengan layanan adalah tipe IAM role unik yang terkait langsung ke layanan. Peran yang terhubung dengan layanan ditentukan sebelumnya oleh AWS DMS dan mencakup semua izin yang diperlukan layanan untuk menghubungi layanan AWS lainnya atas nama Anda.

Peran yang terhubung dengan layanan memudahkan pengaturan AWS DMS karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. AWS DMS menentukan izin peran yang terhubung dengan layanan, dan kecuali ditentukan sebaliknya, hanya AWS DMS yang dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, dan kebijakan izin tersebut tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran terkait layanan hanya setelah menghapus sumber daya yang terkait terlebih dahulu. Ini melindungi sumber daya AWS DMS karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat [AWS Layanan yang Bekerja dengan IAM](#) dan cari layanan yang memiliki Ya di kolom Peran Tertaut Layanan. Pilih Ya dengan tautan untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Peran terkait layanan untuk fitur AWS DMS

Topik

- [Peran terkait layanan untuk AWS DMS Fleet Advisor](#)
- [Peran terkait layanan untuk Tanpa Server AWS DMS](#)

Peran terkait layanan untuk AWS DMS Fleet Advisor

AWS DMS Fleet Advisor menggunakan peran terkait layanan bernama `AWSServiceRoleForDMSFleetAdvisor`— DMS Fleet Advisor menggunakan peran terkait layanan ini untuk mengelola metrik Amazon. CloudWatch Peran terkait layanan ini dilampirkan ke kebijakan terkelola berikut ini: `AWSDMSFleetAdvisorServiceRolePolicy`. Untuk pembaruan kebijakan ini, lihat [Kebijakan terkelola AWS untuk AWS Database Migration Service](#).

`AWSServiceRoleForDMSFleetAdvisor` peran terkait layanan memercayakan layanan berikut untuk menjalankan peran tersebut:

- `dms-fleet-advisor.amazonaws.com`

Kebijakan izin peran bernama `AWSDMSFleetAdvisorServiceRolePolicy` memungkinkan AWS DMS Fleet Advisor untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Tindakan: `cloudwatch:PutMetricData` pada `all AWS resources`

Izin ini memungkinkan prinsipal untuk mempublikasikan titik data metrik ke Amazon. CloudWatch AWS DMS Armada Advisor memerlukan izin ini untuk menampilkan grafik dengan metrik database dari CloudWatch

Contoh kode berikut menunjukkan `AWSDMSFleetAdvisorServiceRolePolicy` kebijakan yang Anda gunakan untuk membuat `AWSDMSFleetAdvisorServiceRolePolicy` peran.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Resource": "*",
    "Action": "cloudwatch:PutMetricData",
    "Condition": {
      "StringEquals": {
        "cloudwatch:namespace": "AWS/DMS/FleetAdvisor"
      }
    }
  }
}
```

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM, seperti pengguna, grup, atau peran, untuk membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat [Izin peran tertaut layanan](#) dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk AWS DMS Fleet Advisor

Anda dapat menggunakan konsol IAM untuk membuat peran terkait layanan dengan kasus penggunaan DMS - Fleet Advisor. Di AWS CLI atau API AWS, buat peran yang terhubung dengan layanan dengan nama layanan `dms-fleet-advisor.amazonaws.com`. Untuk informasi lebih lanjut, lihat [Membuat peran terkait layanan](#) dalam Panduan Pengguna IAM. Jika Anda menghapus peran tertaut layanan ini, Anda dapat mengulang proses yang sama untuk membuat peran tersebut lagi.

Pastikan Anda membuat peran ini sebelum membuat pengumpul data. DMS Armada Advisor menggunakan peran ini untuk menampilkan grafik dengan metrik database di AWS Management Console Untuk informasi selengkapnya, lihat [Membuat pengumpul data](#).

Mengedit peran terkait layanan untuk AWS DMS Fleet Advisor

AWS DMS tidak mengizinkan Anda untuk mengedit peran `AWSServiceRoleForDMSFleetAdvisor` terkait layanan. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat menyunting penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat [Mengedit peran yang terkait dengan layanan](#) dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Fleet Advisor AWS DMS

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan demikian, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Tetapi, Anda harus membersihkan sumber daya peran terkait layanan sebelum menghapusnya secara manual.

Note

Jika layanan AWS DMS menggunakan peran tersebut ketika Anda mencoba menghapus sumber daya, penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus sumber daya AWS DMS yang digunakan oleh `AWSServiceRoleForDMSFleetAdvisor`

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.
2. Di panel navigasi, pilih Pengumpul data di bawah Temukan. Halaman pengumpul data terbuka.
3. Pilih pengumpul data Anda dan pilih Hapus.
4. Untuk mengkonfirmasi penghapusan, masukkan nama pengumpul data di bidang input teks. Selanjutnya, pilih Hapus.

⚠ Important

Ketika Anda menghapus kolektor data DMS, DMS Fleet Advisor menghapus semua database dari Inventory yang Anda temukan menggunakan kolektor ini.

Setelah menghapus semua pengumpul data, Anda dapat menghapus peran terkait layanan.

Untuk menghapus peran tertaut layanan secara manual menggunakan IAM

Gunakan konsol IAM, AWS CLI, atau AWS API untuk menghapus peran terkait layanan `AWSServiceRoleForDMSFleetAdvisor`. Untuk informasi selengkapnya, lihat [Menghapus peran tertaut layanan](#) dalam Panduan Pengguna IAM.

Wilayah yang didukung untuk peran AWS DMS terkait layanan Fleet Advisor

AWS DMSFleet Advisor mendukung penggunaan peran terkait layanan di semua wilayah tempat layanan tersedia. Untuk informasi selengkapnya, lihat [Wilayah AWS yang Didukung](#).

Peran terkait layanan untuk Tanpa Server AWS DMS

AWS DMS Tanpa server menggunakan peran terkait layanan bernama.

`AWSServiceRoleForDMSServerless` AWS DMS menggunakan peran terkait layanan ini untuk membuat dan mengelola AWS DMS sumber daya atas nama Anda, seperti metrik Amazon CloudWatch . AWS DMS menggunakan peran ini sehingga Anda hanya perlu peduli dengan replikasi. Peran terkait layanan ini dilampirkan ke kebijakan terkelola berikut ini: `AWSDMSServerlessServiceRolePolicy`. Untuk pembaruan kebijakan ini, lihat [Kebijakan terkelola AWS untuk AWS Database Migration Service](#).

Peran `AWSServiceRoleForDMSServerless` terkait layanan mempercayai layanan berikut untuk mengambil peran:

- `dms.amazonaws.com`

Contoh kode berikut menunjukkan `AWSDMSServerlessServiceRolePolicy` kebijakan yang Anda gunakan untuk membuat `AWSServiceRoleForDMSServerless` peran.

```
{
  "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:*"
    ]
}
]
```

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM, seperti pengguna, grup, atau peran, untuk membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat [Izin peran terkait layanan](#) dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk Tanpa Server AWS DMS

Saat Anda membuat replikasi, AWS DMS tanpa server secara terprogram membuat peran terkait layanan tanpa server. AWS DMS Anda dapat melihat peran ini di konsol IAM. Anda juga dapat memilih untuk membuat peran ini secara manual. Untuk membuat peran secara manual, gunakan konsol IAM untuk membuat peran terkait layanan dengan kasus penggunaan DMS. Di AWS CLI atau AWS API, buat peran terkait layanan menggunakan nama `dms.amazonaws.com` layanan. Untuk informasi selengkapnya, lihat [Membuat peran terkait layanan di Panduan Pengguna IAM](#). Jika Anda menghapus peran tertaut layanan ini, Anda dapat mengulang proses yang sama untuk membuat peran tersebut lagi.

Note

Jika Anda menghapus peran saat Anda memiliki replikasi di akun Anda, replikasi akan mengakibatkan kegagalan.

Mengedit peran terkait layanan untuk Tanpa Server AWS DMS

AWS DMS tidak memungkinkan Anda untuk mengedit peran `AWSServiceRoleForDMSServerless` terkait layanan. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat menyunting

penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat [Mengedit peran terkait layanan](#) dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Tanpa Server AWS DMS

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan demikian, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Tetapi, Anda harus membersihkan sumber daya peran terkait layanan sebelum menghapusnya secara manual.

Note

Jika AWS DMS layanan menggunakan peran saat Anda mencoba menghapus sumber daya, maka penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus AWS DMS sumber daya yang digunakan oleh `AWSServiceRoleForDMSServerless`

1. Masuk ke AWS Management Console dan buka AWS DMS konsol di <https://console.aws.amazon.com/dms/v2/>.
2. Di panel navigasi, pilih Tanpa Server di bawah Temukan. Halaman Tanpa Server terbuka.
3. Pilih replikasi tanpa server Anda dan pilih Hapus.
4. Untuk mengonfirmasi penghapusan, masukkan nama replikasi tanpa server di bidang input teks. Selanjutnya, pilih Hapus.

Setelah menghapus semua replikasi tanpa server, Anda dapat menghapus peran terkait layanan.

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran `AWSServiceRoleForDMSServerless` terkait layanan. Untuk informasi selengkapnya, lihat [Menghapus peran terkait layanan](#) dalam Panduan Pengguna IAM.

Wilayah yang didukung untuk peran AWS DMS terkait layanan Tanpa Server

AWS DMS Dukungan tanpa server menggunakan peran terkait layanan di semua wilayah tempat layanan tersedia.

Memecahkan masalah AWS Database Migration Service identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS DMS dan IAM.

Topik

- [Saya tidak berwenang untuk melakukan tindakan di AWS DMS](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya seorang administrator dan ingin mengizinkan orang lain mengakses AWS DMS](#)
- [Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses AWS DMS sumber daya saya](#)

Saya tidak berwenang untuk melakukan tindakan di AWS DMS

Jika AWS Management Console memberitahu Anda bahwa Anda tidak berwenang untuk melakukan tindakan, maka Anda harus menghubungi administrator Anda untuk bantuan. Administrator adalah orang yang memberikan nama pengguna dan kata sandi kepada Anda.

Contoh kesalahan berikut terjadi ketika pengguna mateojackson IAM mencoba menggunakan konsol untuk melihat detail tentang titik akhir AWS DMS tetapi tidak memiliki izin. `dms:DescribeEndpoint`

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
dms:DescribeEndpoint on resource: my-postgresql-target
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya agar dia diizinkan mengakses sumber daya titik akhir `my-postgresql-target` menggunakan `dms:DescribeEndpoint` tindakan.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran AWS DMS.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di AWS DMS. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya seorang administrator dan ingin mengizinkan orang lain mengakses AWS DMS

Untuk memungkinkan orang lain mengakses AWS DMS, Anda harus membuat entitas IAM (pengguna atau peran) untuk orang atau aplikasi yang membutuhkan akses. Mereka akan menggunakan kredensial untuk entitas tersebut untuk mengakses AWS. Anda kemudian harus melampirkan kebijakan yang memberi mereka izin yang tepat di AWS DMS.

Untuk segera memulai, lihat [Membuat pengguna dan grup IAM pertama Anda yang didelegasikan](#) di Panduan Pengguna IAM.

Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses AWS DMS sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau pengguna di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi pengguna akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mempelajari apakah AWS DMS mendukung fitur-fitur ini, lihat [Bagaimana AWS Database Migration Service bekerja dengan IAM](#).
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki](#) di Panduan Pengguna IAM.

- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat [Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Memberikan akses kepada pengguna eksternal yang sah \(federasi identitas\)](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara penggunaan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Perbedaan antara peran IAM dan kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

Izin IAM diperlukan untuk menggunakan AWS DMS

Anda menggunakan izin IAM tertentu dan IAM role untuk menggunakan AWS DMS. Jika Anda masuk sebagai pengguna IAM dan ingin menggunakannya AWS DMS, administrator akun Anda harus melampirkan kebijakan yang dibahas di bagian ini ke pengguna, grup, atau peran IAM yang Anda gunakan untuk menjalankan. AWS DMS Untuk informasi lebih lanjut tentang izin IAM, lihat [Panduan Pengguna IAM](#).

Kebijakan berikut memberi Anda akses ke AWS DMS, dan juga izin untuk tindakan tertentu yang diperlukan dari layanan Amazon lainnya seperti AWS KMS, IAM, Amazon EC2, dan Amazon CloudWatch CloudWatch memantau AWS DMS migrasi Anda secara real time dan mengumpulkan serta melacak metrik yang menunjukkan kemajuan migrasi Anda. Anda dapat menggunakan CloudWatch Log untuk men-debug masalah dengan tugas.

Note

Anda selanjutnya dapat membatasi akses ke AWS DMS sumber daya menggunakan penandaan. Untuk informasi selengkapnya tentang membatasi akses ke AWS DMS sumber daya menggunakan penandaan, lihat [Kontrol akses detail menggunakan nama sumber daya dan tag](#)

```
{
  "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"
}
]
}

```

Perincian izin berikut ini dapat membantu Anda lebih memahami mengapa masing-masing izin diperlukan.

Bagian berikut diperlukan untuk memungkinkan pengguna memanggil operasi AWS DMS API.

```
{
    "Effect": "Allow",
    "Action": "dms:*",
    "Resource": "arn:aws:dms:region:account:resourcetype/id"
}
```

Bagian berikut diperlukan untuk memungkinkan pengguna mencantumkan AWS KMS kunci dan alias yang tersedia untuk ditampilkan di konsol. Entri ini tidak diperlukan jika Anda mengetahui Nama Sumber Daya Amazon (ARN) untuk kunci KMS dan Anda hanya menggunakan (). AWS Command Line Interface AWS CLI

```
{
    "Effect": "Allow",
    "Action": [
        "kms:ListAliases",
        "kms:DescribeKey"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}
```

Bagian berikut ini diperlukan untuk jenis titik akhir tertentu yang memerlukan ARN IAM role untuk diteruskan dengan titik akhir. Selain itu, jika AWS DMS peran yang diperlukan tidak dibuat sebelumnya, AWS DMS konsol dapat membuat peran. Jika semua peran dikonfigurasi sebelumnya, yang diperlukan adalah `iam:GetRole` dan `iam:PassRole`. Untuk informasi lebih lanjut tentang peran, lihat [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#).

```
{
    "Effect": "Allow",
    "Action": [
        "iam:GetRole",
        "iam:PassRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy"
    ]
}
```

```

    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
  }

```

Bagian berikut diperlukan karena AWS DMS perlu membuat instans Amazon EC2 dan mengonfigurasi jaringan untuk instance replikasi yang dibuat. Sumber daya ini ada di akun pelanggan, sehingga kemampuan untuk melakukan tindakan ini atas nama pelanggan diperlukan.

```

{
  "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"
}

```

Bagian berikut diperlukan untuk mengizinkan pengguna untuk dapat melihat metrik instans replikasi.

```

{
  "Effect": "Allow",
  "Action": [
    "cloudwatch:Get*",
    "cloudwatch:List*"
  ],
  "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

Bagian ini diperlukan untuk mengizinkan pengguna untuk melihat log replikasi.

```

{
  "Effect": "Allow",
  "Action": [
    "logs:DescribeLogGroups",
    "logs:DescribeLogStreams",
    "logs:FilterLogEvents",

```

```
        "logs:GetLogEvents"  
    ],  
    "Resource": "arn:aws:service:region:account:resourcetype/id"  
}
```

AWS DMS Konsol membuat beberapa peran yang secara otomatis dilampirkan ke AWS akun Anda saat Anda menggunakan AWS DMS konsol. Jika Anda menggunakan AWS Command Line Interface (AWS CLI) atau AWS DMS API untuk migrasi, Anda perlu menambahkan peran ini ke akun Anda. Untuk informasi lebih lanjut tentang menambahkan peran ini, lihat [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#).

Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS

Jika Anda menggunakan AWS CLI atau AWS DMS API untuk migrasi database, Anda harus menambahkan tiga peran IAM ke AWS akun Anda sebelum dapat menggunakan fitur dari AWS DMS. Dua dari peran tersebut adalah `dms-vpc-role` dan `dms-cloudwatch-logs-role`. Jika Anda menggunakan Amazon Redshift sebagai database target, Anda juga harus menambahkan peran IAM `dms-access-for-endpoint` ke akun Anda. AWS

Pembaruan untuk kebijakan terkelola otomatis. Jika Anda menggunakan kebijakan kustom dengan IAM role, pastikan untuk secara berkala memeriksa pembaruan untuk kebijakan terkelola dalam dokumentasi ini. Anda dapat melihat detail kebijakan terkelola dengan menggunakan kombinasi perintah `get-policy` dan `get-policy-version`.

Sebagai contoh, perintah `get-policy` mengambil informasi tentang IAM role yang ditentukan.

```
aws iam get-policy --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonDMSVPCManagementRole
```

Informasi yang dikembalikan dari perintah adalah sebagai berikut.

```
{
  "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": "ANPAJHKIGMBQI4AEFFSY0",
    "DefaultVersionId": "v3",
    "Path": "/service-role/",
    "Arn": "arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole",
    "UpdateDate": "2016-05-23T16:29:57Z"
  }
}
```

Perintah `get-policy-version` berikut mengambil informasi kebijakan IAM.

```
aws iam get-policy-version --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonDMSVPCManagementRole --version-id v3
```

Informasi yang dikembalikan dari perintah adalah sebagai berikut.

```
{
  "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
  }
}
```

Anda dapat menggunakan perintah yang sama untuk mendapatkan informasi tentang kebijakan terkelola AmazonDMSCloudWatchLogsRole dan AmazonDMSRedshiftS3Role.

Note

Jika Anda menggunakan AWS DMS konsol untuk migrasi database, peran ini akan ditambahkan ke AWS akun Anda secara otomatis.

Prosedur berikut membuat IAM role `dms-vpc-role`, `dms-cloudwatch-logs-role`, dan `dms-access-for-endpoint`.

Untuk membuat peran `dms-vpc-role` IAM untuk digunakan dengan AWS CLI atau API AWS DMS

1. Membuat file JSON dengan kebijakan IAM berikut. Beri nama file JSON `dmsAssumeRolePolicyDocument.json`.

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

Buat peran menggunakan AWS CLI menggunakan perintah berikut.

```
aws iam create-role --role-name dms-vpc-role --assume-role-policy-document file://
dmsAssumeRolePolicyDocument.json
```

2. Lampirkan kebijakan `AmazonDMSVPCManagementRole` ke `dms-vpc-role` menggunakan perintah berikut ini.

```
aws iam attach-role-policy --role-name dms-vpc-role --policy-arn
arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole
```

Untuk membuat peran `dms-cloudwatch-logs-role` IAM untuk digunakan dengan AWS CLI atau API AWS DMS

1. Membuat file JSON dengan kebijakan IAM berikut. Beri nama file JSON `dmsAssumeRolePolicyDocument2.json`.

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

Buat peran menggunakan AWS CLI menggunakan perintah berikut.

```
aws iam create-role --role-name dms-cloudwatch-logs-role --assume-role-policy-
document file://dmsAssumeRolePolicyDocument2.json
```

2. Lampirkan kebijakan `AmazonDMSCloudWatchLogsRole` ke `dms-cloudwatch-logs-role` menggunakan perintah berikut ini.

```
aws iam attach-role-policy --role-name dms-cloudwatch-logs-role --policy-arn
arn:aws:iam::aws:policy/service-role/AmazonDMSCloudWatchLogsRole
```

Jika Anda menggunakan Amazon Redshift sebagai basis data target Anda, Anda harus membuat IAM role `dms-access-for-endpoint` untuk menyediakan akses ke Amazon S3.

Untuk membuat peran `dms-access-for-endpoint` IAM untuk digunakan dengan Amazon Redshift sebagai database target

1. Membuat file JSON dengan kebijakan IAM berikut. Beri nama file 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. Buat peran menggunakan AWS CLI menggunakan perintah berikut.

```
aws iam create-role --role-name dms-access-for-endpoint --assume-role-policy-document file://dmsAssumeRolePolicyDocument3.json
```

-
-
3. Lampirkan kebijakan AmazonDMSRedshiftS3Role ke peran `dms-access-for-endpoint` menggunakan perintah berikut ini.

```
aws iam attach-role-policy --role-name dms-access-for-endpoint \  
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonDMSRedshiftS3Role
```

Anda sekarang harus memiliki kebijakan IAM untuk menggunakan AWS CLI atau AWS DMS API.

Pencegahan confused deputy lintas layanan

Masalah confused deputy adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang lebih berhak untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan pengguna utama layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan dalam kebijakan sumber daya untuk membatasi izin yang AWS Database Migration Service memberikan layanan lain ke sumber daya. Jika `aws:SourceArn` nilai tidak berisi ID akun, seperti nama instance AWS DMS replikasi (ARN), Anda harus menggunakan kedua kunci konteks kondisi global untuk membatasi izin. Jika Anda menggunakan kunci konteks kondisi global dan nilai `aws:SourceArn` berisi ID akun, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama. Gunakan `aws:SourceArn` jika Anda hanya ingin satu sumber daya dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

AWS DMS mendukung opsi wakil yang bingung mulai dari versi 3.4.7 dan lebih tinggi. Untuk informasi selengkapnya, lihat [AWS Catatan rilis Database Migration Service 3.4.7](#). Jika instans replikasi Anda menggunakan AWS DMS versi 3.4.6 atau yang lebih rendah, pastikan Anda meningkatkan ke versi terbaru sebelum menetapkan opsi deputy yang membingungkan.

Cara paling efektif untuk melindungi dari masalah confused deputy adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan karakter wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:dms:*:123456789012:rep:*`.

Topik

- [Peran IAM untuk digunakan dengan AWS DMS API untuk pencegahan wakil yang membingungkan lintas layanan](#)

- [Kebijakan IAM untuk menyimpan penilaian preflight di Amazon S3 untuk pencegahan wakil yang membingungkan lintas layanan](#)
- [Menggunakan Amazon DynamoDB sebagai titik akhir target AWS DMS dengan pencegahan wakil kebingungan lintas layanan](#)

Peran IAM untuk digunakan dengan AWS DMS API untuk pencegahan wakil yang membingungkan lintas layanan

Untuk menggunakan AWS CLI atau AWS DMS API untuk migrasi database, Anda harus menambahkan peran `dms-vpc-role` dan `dms-cloudwatch-logs-role` IAM ke AWS akun Anda sebelum dapat menggunakan fitur AWS DMS. Untuk informasi selengkapnya, lihat [Membuat peran IAM untuk digunakan dengan AWS CLI dan API AWS DMS](#).

Contoh berikut menunjukkan kebijakan untuk menggunakan `dms-vpc-role` peran dengan contoh `my-replication-instance` replikasi. Gunakan kebijakan ini untuk mencegah masalah wakil yang membingungkan.

```
{
  "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"
        }
      }
    }
  ]
}
```

Kebijakan IAM untuk menyimpan penilaian preflight di Amazon S3 untuk pencegahan wakil yang membingungkan lintas layanan

Untuk menyimpan hasil prasesmen di bucket S3, Anda membuat kebijakan IAM yang memungkinkan AWS DMS untuk mengelola objek di Amazon S3. Untuk informasi selengkapnya, lihat [Buat sumber daya IAM](#).

Contoh berikut menunjukkan kebijakan kepercayaan dengan kondisi wakil yang membingungkan yang ditetapkan pada peran IAM yang memungkinkan AWS DMS untuk mengakses semua tugas dan penilaian berjalan di bawah akun pengguna tertentu.

```
{
  "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:*"
          ]
        }
      }
    }
  ]
}
```

Menggunakan Amazon DynamoDB sebagai titik akhir target AWS DMS dengan pencegahan wakil kebingungan lintas layanan

Untuk menggunakan Amazon DynamoDB sebagai titik akhir target untuk migrasi database, Anda harus membuat peran IAM yang AWS DMS memungkinkan untuk mengasumsikan dan memberikan

akses ke tabel DynamoDB. Kemudian, gunakan peran ini saat Anda membuat titik akhir DynamoDB target di. AWS DMS Untuk informasi selengkapnya, lihat [Menggunakan Amazon DynamoDB sebagai target](#).

Contoh berikut menunjukkan kebijakan kepercayaan dengan kondisi deputi bingung yang ditetapkan pada peran IAM yang memungkinkan semua AWS DMS titik akhir mengakses tabel 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:*"
        }
      }
    }
  ]
}
```

Kebijakan terkelola AWS untuk AWS Database Migration Service

Topik

- [Kebijakan terkelola AWS: AWSDMSServerlessServiceRolePolicy](#)
- [AWSkebijakan terkelola: AmazonDMS CloudWatchLogsRole](#)
- [Kebijakan terkelola AWS: AWSDMSFleetAdvisorServiceRolePolicy](#)
- [AWS DMS memperbarui pada kebijakan terkelola AWS](#)

Kebijakan terkelola AWS: AWSDMSServerlessServiceRolePolicy

Kebijakan ini melekat pada `AWSServiceRoleForDMSServerless` peran, yang memungkinkan AWS DMS untuk melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat [Peran terkait layanan untuk Tanpa Server AWS DMS](#).

Kebijakan ini memberikan izin kontributor yang memungkinkan AWS DMS untuk mengelola sumber daya replikasi.

Rincian perizinan

Kebijakan ini mencakup izin berikut.

- `dms-` Memungkinkan prinsipal untuk berinteraksi dengan sumber daya. 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:*"
    ]
}
]
}

```

AWSkebijakan terkelola: AmazonDMS CloudWatchLogsRole

Kebijakan ini melekat pada `dms-cloudwatch-logs-role` peran, yang memungkinkan AWS DMS untuk melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat [Menggunakan peran terkait layanan untuk AWS DMS](#).

Kebijakan ini memberikan izin kontributor yang memungkinkan AWS DMS untuk mempublikasikan log replikasi ke log. CloudWatch

Rincian perizinan

Kebijakan ini mencakup izin berikut.

- logs- Memungkinkan prinsipal untuk mempublikasikan log ke Log. CloudWatch Izin ini diperlukan sehingga AWS DMS dapat digunakan CloudWatch untuk menampilkan log replikasi.

```
{
  "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-*"
      ]
    }
  ]
}

```

Kebijakan terkelola AWS: AWSDMSFleetAdvisorServiceRolePolicy

Anda tidak dapat melampirkan AWSDMSFleetAdvisorServiceRolePolicy ke entitas IAM Anda. Kebijakan ini dilampirkan pada peran terkait layanan yang memungkinkan AWS DMS Fleet Advisor melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat [Menggunakan peran terkait layanan untuk AWS DMS](#).

Kebijakan ini memberikan izin kontributor yang memungkinkan AWS DMS Fleet Advisor mempublikasikan metrik Amazon. CloudWatch

Rincian perizinan

Kebijakan ini mencakup izin berikut.

- `cloudwatch`- Memungkinkan prinsipal untuk mempublikasikan titik data metrik ke Amazon. CloudWatch Izin ini diperlukan agar AWS DMS Fleet Advisor dapat digunakan CloudWatch untuk menampilkan grafik dengan metrik database.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Resource": "*",
    "Action": "cloudwatch:PutMetricData",
    "Condition": {
      "StringEquals": {
        "cloudwatch:namespace": "AWS/DMS/FleetAdvisor"
      }
    }
  }
}
```

AWS DMS memperbarui pada kebijakan terkelola AWS

Lihat detail tentang pembaruan terhadap kebijakan terkelola AWS untuk AWS DMS sejak layanan ini mulai melacak perubahan-perubahan tersebut. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman Riwayat dokumen AWS DMS.

Perubahan	Deskripsi	Tanggal
AWSDMSServerlessServiceRolePolicy – Kebijakan baru	AWS DMS menambahkan <code>AWSDMSServerlessServiceRolePolicy</code> peran AWS DMS untuk memungkinkan membuat dan mengelola layanan atas nama Anda, seperti menerbitkan CloudWatch metrik Amazon.	22 Mei 2023

Perubahan	Deskripsi	Tanggal
AmazonDMS CloudWatchLogsRole - Perubahan	AWS DMS menambahkan ARN untuk sumber daya tanpa server ke setiap izin yang diberikan, untuk memungkinkan mengunggah log replikasi dari konfigurasi AWS DMS replikasi tanpa server ke Log. CloudWatch	22 Mei 2023
AWS DMS Fleet Advisor Service Role Policy – Kebijakan baru	AWS DMS Fleet Advisor menambahkan kebijakan baru untuk memungkinkan penerbitan poin data metrik ke Amazon. CloudWatch	6 Maret 2023
AWS DMS mulai melacak perubahan	AWS DMS mulai melacak perubahan untuk kebijakan terkelola AWS	6 Maret 2023

Validasi kepatuhan untuk AWS Database Migration Service

Auditor pihak ke tiga menilai keamanan dan kepatuhan AWS Database Migration Service sebagai bagian dari beberapa program kepatuhan AWS. Hal yang dimaksud meliputi program berikut:

- SOC
- PCI
- ISO
- FedRAMP
- DoD CC SRG
- HIPAA BAA
- MTCS
- CS
- K-ISMS
- ENS High
- OSPAR
- HITRUST CSF

Untuk daftar layanan AWS dalam cakupan program kepatuhan tertentu, lihat [Layanan AWS dalam cakupan melalui program kepatuhan](#). Untuk informasi umum, lihat [AWS program kepatuhan](#).

Anda bisa mengunduh laporan audit pihak ke tiga menggunakan AWS Artifact. Untuk informasi lebih lanjut, lihat [Pengunduhan laporan di AWS artifact](#).

Tanggung jawab kepatuhan Anda saat menggunakan AWS DMS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta undang-undang dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu dengan kepatuhan:

- [Panduan quick start keamanan dan kepatuhan](#) – Panduan deployment ini membahas pertimbangan arsitektur dan memberikan langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan di AWS.
- [Laporan resmi perancangan untuk keamanan dan kepatuhan HIPAA di Amazon Web Services](#) — Laporan resmi ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang mematuhi HIPAA.

- [Sumber daya kepatuhanAWS](#) – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Config](#) – Layanan AWS ini menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#) – Layanan AWS ini akan menyediakan tampilan komprehensif dari status keamanan Anda dalam AWS yang akan membantu Anda memeriksa kepatuhan Anda terhadap standar dan praktik terbaik industri.

Ketahanan di AWS Database Migration Service

Infrastruktur global AWS dibangun di sekitar Wilayah dan Availability Zone AWS. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah dan terisolasi secara fisik, yang terhubung dengan jaringan yang memiliki latensi rendah, throughput tinggi, dan sangat berkelebihan. Dengan Availability Zone, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara Availability Zone tanpa gangguan. Availability Zone memiliki ketersediaan yang lebih baik, menoleransi kegagalan, dan dapat diskalakan dibandingkan satu atau beberapa infrastruktur pusat data tradisional.

Untuk informasi lebih lanjut tentang Wilayah AWS dan Availability Zone, lihat [Infrastruktur global AWS](#).

Selain infrastruktur global AWS, AWS DMS menyediakan ketersediaan tinggi dan dukungan failover untuk instans replikasi menggunakan deployment Multi-AZ saat Anda memilih opsi Multi-AZ.

Dalam deployment Multi-AZ, AWS DMS secara otomatis menyediakan dan mempertahankan replika siaga instans replikasi di Availability Zone yang berbeda. Instans replikasi primer direplikasi secara sinkron ke replika siaga. Jika instans replikasi utama gagal atau menjadi tidak responsif, cadangan tersebut melanjutkan tugas yang sedang berlangsung dengan gangguan minimal. Karena instans replikasi utama terus-menerus mereplikasi keadaan pada cadangannya, deployment Multi-AZ tidak dikenakan beberapa overhead performa.

Untuk informasi lebih lanjut tentang bekerja dengan deployment Multi-AZ, lihat [Bekerja dengan instance AWS DMS replikasi](#).

Keamanan infrastruktur dalam AWS Database Migration Service

Sebagai layanan terkelola, AWS Database Migration Service dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat [Keamanan AWS Cloud](#). Untuk merancang AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan](#) Infrastruktur dalam Kerangka Kerja Pilar Keamanan yang AWS Diarsiteksikan dengan Baik.

Anda menggunakan panggilan API AWS yang dipublikasikan untuk mengakses AWS DMS melalui jaringan. Klien harus mendukung hal berikut:

- Transport Layer Security (TLS). Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Suite cipher dengan kerahasiaan maju sempurna (PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan sistem yang lebih baru mendukung mode ini.

Selain itu, permintaan harus ditandatangani menggunakan access key ID dan secret access key yang terkait dengan principal IAM. Atau Anda bisa menggunakan [AWS Security Token Service](#) (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

Anda dapat menghubungi operasi API ini dari lokasi jaringan mana pun. AWS DMS juga mendukung kebijakan akses berbasis sumber daya, yang dapat menentukan pembatasan terhadap tindakan dan sumber daya, misalnya, berdasarkan alamat IP sumber. Selain itu, Anda dapat menggunakan kebijakan AWS DMS untuk mengontrol akses dari Amazon VPC endpoint tertentu atau virtual private cloud (VPC) tertentu. Secara efektif, ini mengisolasi akses jaringan ke sumber daya AWS DMS tertentu hanya dari VPC tertentu dalam jaringan AWS. Untuk informasi lebih lanjut tentang penggunaan kebijakan akses berbasis sumber daya dengan AWS DMS, termasuk contoh, lihat [Kontrol akses detail menggunakan nama sumber daya dan tag](#).

Untuk membatasi komunikasi Anda dengan AWS DMS dalam satu VPC, Anda dapat membuat titik akhir antarmuka VPC yang memungkinkan Anda untuk terhubung ke AWS DMS melalui AWS PrivateLink. AWS PrivateLink membantu memastikan bahwa setiap panggilan ke AWS DMS dan hasil yang terkait tetap terbatas pada VPC tertentu yang titik akhir antarmuka-nya Anda buat. Anda kemudian dapat menentukan URL untuk titik akhir antarmuka ini sebagai opsi dengan setiap perintah AWS DMS yang Anda jalankan menggunakan AWS CLI atau SDK. Melakukan hal ini akan membantu memastikan bahwa seluruh komunikasi Anda dengan AWS DMS tetap terbatas pada VPC dan tidak terlihat oleh internet publik.

Untuk membuat titik akhir antarmuka untuk mengakses DMS di VPC tunggal

1. Masuk ke AWS Management Console dan buka konsol Amazon VPC di <https://console.aws.amazon.com/vpc/>.
2. Dari panel navigasi, pilih Titik akhir. Ini akan membuka halaman Buat titik akhir, di mana Anda dapat membuat titik akhir antarmuka dari VPC ke AWS DMS.
3. Pilih Layanan AWS, lalu cari dan pilih nilai untuk Nama Layanan, dalam hal ini AWS DMS dalam formulir berikut.

```
com.amazonaws.region.dms
```

Di sini, *region* menentukan Wilayah AWS di mana AWS DMS berjalan, misalnya `com.amazonaws.us-west-2.dms`.

4. Untuk VPC, pilih VPC untuk membuat titik akhir antarmuka, misalnya `vpc-12abcd34`.
5. Pilih nilai untuk Availability Zone dan untuk ID subnet. Nilai-nilai ini harus menunjukkan lokasi di mana titik akhir AWS DMS Anda dapat berjalan, misalnya `us-west-2a` (`usw2-az1`) dan `subnet-ab123cd4`.
6. Pilih Aktifkan nama DNS untuk membuat titik akhir dengan nama DNS. Nama DNS ini terdiri dari ID titik akhir (`vpce-12abcd34efg567hij`) yang ditulis dengan tanda hubung dengan string acak (`ab12dc34`). Ini dipisahkan dari nama layanan oleh sebuah titik dalam urutan yang dipisahkan titik terbalik, dengan penambahan `vpce` (`dms.us-west-2.vpce.amazonaws.com`).

Contohnya `vpce-12abcd34efg567hij-ab12dc34.dms.us-west-2.vpce.amazonaws.com`.

7. Untuk Grup keamanan, pilih grup yang akan digunakan untuk titik akhir.

Saat Anda mengatur grup keamanan, pastikan untuk mengizinkan panggilan HTTPS ke luar dari dalamnya. Untuk informasi lebih lanjut, lihat [Membuat grup keamanan](#) di Panduan Pengguna Amazon VPC.

8. Pilih Akses Penuh atau nilai kustom untuk Kebijakan. Misalnya, Anda mungkin memilih kebijakan kustom yang serupa dengan kebijakan berikut ini yang membatasi akses titik akhir Anda ke tindakan dan sumber daya tertentu.

```
{  
  "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": "*"
}
]
```

Di sini, kebijakan sampel memungkinkan panggilan AWS DMS API, kecuali untuk menghapus atau memodifikasi instans replikasi tertentu.

Sekarang Anda dapat menentukan URL yang dibentuk menggunakan nama DNS yang dibuat pada langkah 6 sebagai sebuah opsi. Anda menentukan hal ini untuk setiap perintah AWS DMS CLI atau operasi API untuk mengakses instans layanan menggunakan titik akhir antarmuka yang dibuat. Misalnya, Anda dapat menjalankan perintah DMS CLI `DescribeEndpoints` dalam VPC ini seperti yang ditunjukkan berikut ini.

```
$ aws dms describe-endpoints --endpoint-url https://vpce-12abcd34efg567hij-ab12dc34.dms.us-west-2.vpce.amazonaws.com
```

Jika Anda mengaktifkan opsi DNS privat, Anda tidak perlu menentukan URL titik akhir dalam permintaan.

Untuk informasi selengkapnya tentang membuat dan menggunakan titik akhir antarmuka VPC (termasuk mengaktifkan opsi DNS pribadi), lihat [Endpoint \(AWSPrivateLink\) Antarmuka VPC di Panduan Pengguna Amazon VPC](#).

Kontrol akses detail menggunakan nama sumber daya dan tag

Anda dapat menggunakan nama sumber daya dan tag sumber daya berdasarkan Amazon Resource Name (ARN) untuk mengelola akses ke sumber daya AWS DMS. Anda melakukan ini dengan mendefinisikan tindakan yang diizinkan atau memasukkan pernyataan bersyarat dalam kebijakan IAM.

Menggunakan nama sumber daya untuk mengontrol akses

Anda dapat membuat akun pengguna IAM dan menetapkan kebijakan berdasarkan ARN sumber daya AWS DMS.

Kebijakan berikut menolak akses ke instans replikasi AWS DMS dengan 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"
    }
  ]
}
```

Sebagai contoh, perintah berikut gagal ketika kebijakan berlaku.

```
$ 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:D0H67ZT0XGLIXMIHKITV"
```

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:D0H67ZT0XGLIXMIHKITV

Anda juga dapat menentukan kebijakan IAM yang membatasi akses ke titik akhir AWS DMS dan tugas replikasi.

Kebijakan berikut membatasi akses ke titik akhir AWS DMS menggunakan ARN titik akhir tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:dms:us-
east-1:152683116:endpoint:D6E37YBXTNH0A6XRQSZCUGX"
    }
  ]
}
```

Sebagai contoh, perintah berikut gagal ketika kebijakan menggunakan ARN titik akhir tersebut berlaku.

```
$ aws dms delete-endpoint
  --endpoint-arn "arn:aws:dms:us-east-1:152683116:endpoint:D6E37YBXTNH0A6XRQSZCUGX"
```

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:D6E37YBXTNH0A6XRQSZCUGX

$ aws dms modify-endpoint
  --endpoint-arn "arn:aws:dms:us-east-1:152683116:endpoint:D6E37YBXTNH0A6XRQSZCUGX"
```

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:D6E37YBXTNH0A6XRQSZCUGX
```

Kebijakan berikut membatasi akses ke tugas AWS DMS menggunakan ARN tugas tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:dms:us-east-1:152683116:task:U03YR4N47DXH3ATT4YMW0IT"
    }
  ]
}
```

Sebagai contoh, perintah berikut gagal ketika kebijakan menggunakan ARN tugas tersebut berlaku.

```
$ aws dms delete-replication-task
  --replication-task-arn "arn:aws:dms:us-east-1:152683116:task:U03YR4N47DXH3ATT4YMW0IT"
```

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:U03YR4N47DXH3ATT4YMW0IT
```

Menggunakan tag untuk mengontrol akses

AWS DMS mendefinisikan serangkaian pasangan nilai kunci umum yang tersedia untuk digunakan dalam kebijakan yang ditetapkan pelanggan tanpa persyaratan penandaan tambahan. Untuk informasi lebih lanjut tentang penandaan sumber daya AWS DMS, lihat [Penandaan sumber daya di AWS Database Migration Service](#).

Daftar berikut ini mencantumkan tag standar yang tersedia untuk digunakan dengan AWS DMS:

- `aws: CurrentTime` — Merupakan tanggal dan waktu permintaan, memungkinkan pembatasan akses berdasarkan kriteria temporal.
- `aws: EpochTime` — Tag ini mirip dengan `CurrentTime` tag `aws:` sebelumnya, kecuali bahwa waktu saat ini direpresentasikan sebagai jumlah detik yang telah berlalu sejak zaman Unix.
- `aws: MultiFactorAuthPresent` — Ini adalah tag Boolean yang menunjukkan apakah permintaan ditandatangani melalui otentikasi multi-faktor atau tidak.
- `aws: MultiFactorAuthAge` — Menyediakan akses ke usia token otentikasi multi-faktor (dalam hitungan detik).
- `aws: principaltype` - Menyediakan akses ke jenis prinsipal (pengguna, akun, pengguna gabungan, dll) untuk permintaan saat ini.
- `aws: SourceIp` — Merupakan alamat ip sumber untuk pengguna yang mengeluarkan permintaan.
- `aws: UserAgent` — Menyediakan informasi tentang aplikasi klien yang meminta sumber daya.
- `aws:userid` - Menyediakan akses ke ID dari pengguna yang mengeluarkan permintaan.
- `aws:username` - Menyediakan akses ke nama pengguna yang mengeluarkan permintaan.
- `dms: InstanceClass` — Menyediakan akses ke ukuran komputasi host instance replikasi.
- `dms: StorageSize` — Menyediakan akses ke ukuran volume penyimpanan (dalam GB).

Anda juga dapat menentukan tag Anda sendiri. Tag yang ditetapkan pelanggan adalah pasangan nilai kunci sederhana yang bertahan dalam layanan penandaan AWS. Anda dapat menambahkan ini ke sumber daya AWS DMS, termasuk instans replikasi, titik akhir, dan tugas. Tag ini dicocokkan dengan menggunakan pernyataan "bersyarat" IAM dalam kebijakan, dan direferensikan menggunakan tag bersyarat tertentu. Kunci tag diawali dengan "dms", jenis sumber daya, dan prefiks "tag". Berikut ini menunjukkan format tag.

```
dms:{resource type}-tag/{tag key}={tag value}
```

Sebagai contoh, misalkan Anda ingin menentukan kebijakan yang hanya mengizinkan panggilan API berhasil untuk instans replikasi yang berisi tag "stage=production". Pernyataan bersyarat berikut cocok dengan sumber daya dengan tag yang diberikan.

```
"Condition":
{
  "streq":
  {
    "dms:rep-tag/stage":"production"
  }
}
```

Anda menambahkan tag berikut ke instans replikasi yang cocok dengan syarat kebijakan ini.

```
stage production
```

Selain tag yang sudah ditugaskan untuk sumber daya AWS DMS, kebijakan juga dapat ditulis untuk membatasi kunci dan nilai tag yang dapat diterapkan ke sumber daya tertentu. Dalam hal ini, prefiks tag adalah "req".

Sebagai contoh, pernyataan kebijakan berikut membatasi tag yang dapat ditetapkan pengguna ke sumber daya tertentu untuk daftar tertentu dari nilai yang diizinkan.

```
"Condition":
{
  "streq":
  {
    "dms:rep-tag/stage": [ "production", "development", "testing" ]
  }
}
```

Contoh kebijakan berikut membatasi akses ke sumber daya AWS DMS berdasarkan tag sumber daya.

Kebijakan berikut membatasi akses ke instans replikasi di mana nilai tag adalah "Desktop" dan kunci tag adalah "Env":

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "dms:rep-tag/Env": [
            "Desktop"
          ]
        }
      }
    }
  ]
}
```

Perintah berikut berhasil atau gagal berdasarkan kebijakan IAM yang membatasi akses ketika nilai tag adalah "Desktop" dan kunci tag adalah "Env".

```
$ aws dms list-tags-for-resource
--resource-name arn:aws:dms:us-east-1:152683116:rep:46DH0U7J0JY0JXWDOZNFEN
--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:46DH0U7J0JY0JXWDOZNFEN"
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:46DHOU7J0JY0JXWDOZNFEN
```

```
$ aws dms modify-replication-instance
  --replication-instance-arn "arn:aws:dms:us-
east-1:152683116:rep:46DHOU7J0JY0JXWDOZNFEN"
```

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:46DHOU7J0JY0JXWDOZNFEN
```

```
$ aws dms add-tags-to-resource
  --resource-name arn:aws:dms:us-east-1:152683116:rep:46DHOU7J0JY0JXWDOZNFEN
  --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:46DHOU7J0JY0JXWDOZNFEN

```
$ aws dms remove-tags-from-resource
  --resource-name arn:aws:dms:us-east-1:152683116:rep:46DHOU7J0JY0JXWDOZNFEN
  --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:46DHOU7J0JY0JXWDOZNFEN
```

Kebijakan berikut membatasi akses ke titik akhir AWS DMS di mana nilai tag adalah "Desktop" dan kunci tag adalah "Env".

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
```

```

        "dms:*"
    ],
    "Effect": "Deny",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "dms:endpoint-tag/Env": [
                "Desktop"
            ]
        }
    }
}
]
}

```

Perintah berikut berhasil atau gagal berdasarkan kebijakan IAM yang membatasi akses ketika nilai tag adalah "Desktop" dan kunci tag adalah "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:J2YCZPNGOLF52344IZWA6I

```
$ aws dms add-tags-to-resource
  --resource-name arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLF52344IZWA6I
  --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:J2YCZPNGOLF52344IZWA6I

```
$ aws dms remove-tags-from-resource
  --resource-name arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLF52344IZWA6I
  --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:J2YCZPNGOLF52344IZWA6I

Kebijakan berikut membatasi akses ke tugas replikasi di mana nilai tag adalah "Desktop" dan kunci tag adalah "Env".

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "dms:task-tag/Env": [
            "Desktop"
          ]
        }
      }
    }
  ]
}
```

```

    }
  }
]
}

```

Perintah berikut berhasil atau gagal berdasarkan kebijakan IAM yang membatasi akses ketika nilai tag adalah "Desktop" dan kunci tag adalah "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

Mengatur kunci enkripsi dan menentukan izin AWS KMS

AWS DMS mengenkripsi penyimpanan yang digunakan oleh instans replikasi dan informasi koneksi titik akhir. Untuk mengenkripsi penyimpanan yang digunakan oleh instans replikasi, AWS DMS menggunakan kunci AWS Key Management Service (AWS KMS) yang unik untuk akun AWS Anda. Anda dapat melihat dan mengelola kunci ini dengan AWS KMS. Anda dapat menggunakan kunci KMS default di akun Anda (`aws/dms`) atau Anda dapat membuat kunci KMS khusus. Jika Anda memiliki kunci KMS yang ada, Anda juga dapat menggunakan kunci itu untuk enkripsi.

Note

Setiap kunci AWS KMS kustom atau kunci yang sudah ada yang Anda gunakan sebagai kunci enkripsi harus kunci simetris. AWS DMS tidak mendukung penggunaan kunci enkripsi asimetris. Untuk informasi lebih lanjut tentang kunci enkripsi simetris dan asimetris, lihat <https://docs.aws.amazon.com/kms/latest/developerguide/symmetric-asymmetric.html> di Panduan Developer AWS Key Management Service.

Kunci KMS default (`aws/dms`) dibuat saat pertama kali meluncurkan instance replikasi, jika Anda belum memilih kunci KMS kustom dari bagian Advanced pada halaman Create Replication Instance. Jika Anda menggunakan kunci KMS default, satu-satunya izin yang perlu Anda berikan ke akun pengguna IAM yang Anda gunakan untuk migrasi adalah `kms:ListAliases` dan `kms:DescribeKey`. Untuk informasi selengkapnya tentang menggunakan kunci KMS default, lihat [izin IAM diperlukan untuk menggunakan AWS DMS](#).

Untuk menggunakan kunci KMS kustom, tetapkan izin untuk kunci KMS kustom menggunakan salah satu opsi berikut:

- Tambahkan akun pengguna IAM yang digunakan untuk migrasi sebagai administrator kunci atau pengguna kunci untuk kunci kustom AWS KMS. Melakukan hal ini memastikan bahwa izin AWS KMS yang diperlukan diberikan ke akun pengguna IAM. Tindakan ini sebagai tambahan untuk izin IAM yang Anda berikan ke akun pengguna IAM untuk menggunakan AWS DMS. Untuk informasi selengkapnya tentang pemberian izin kepada pengguna kunci, lihat [Mengizinkan pengguna kunci menggunakan kunci KMS di Panduan Pengembang](#) AWS Key Management Service.
- Jika Anda tidak ingin menambahkan akun pengguna IAM sebagai administrator kunci atau pengguna kunci untuk kunci KMS kustom Anda, tambahkan izin tambahan berikut ke izin IAM yang harus Anda berikan ke akun pengguna IAM untuk digunakan. AWS DMS

```
{
    "Effect": "Allow",
    "Action": [
        "kms:ListAliases",
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:Encrypt",
        "kms:ReEncrypt*"
    ],
    "Resource": "*"
},
```

AWS DMS juga bekerja dengan alias kunci KMS. Untuk informasi selengkapnya tentang membuat AWS KMS kunci Anda sendiri dan memberi pengguna akses ke kunci KMS, lihat [Panduan AWS KMS Pengembang](#).

Jika Anda tidak menentukan pengenalan kunci KMS, maka AWS DMS gunakan kunci enkripsi default Anda. AWS KMS membuat kunci enkripsi default AWS DMS untuk AWS akun Anda. Akun AWS Anda memiliki kunci enkripsi default yang berbeda untuk setiap Wilayah AWS.

Untuk mengelola kunci AWS KMS yang digunakan untuk mengenkripsi sumber daya AWS DMS Anda, gunakan AWS Key Management Service. AWS KMS menggabungkan perangkat keras dan perangkat lunak yang aman dan selalu tersedia untuk menyediakan sistem manajemen kunci yang diskalakan untuk cloud. Menggunakan AWS KMS, Anda dapat membuat kunci enkripsi dan menetapkan kebijakan yang mengendalikan bagaimana kunci tersebut digunakan.

Anda dapat menemukan AWS KMS di AWS Management Console

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di <https://console.aws.amazon.com/kms>.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Pilih salah satu opsi berikut untuk menggunakan kunci AWS KMS:
 - Untuk melihat tombol di akun yang dibuat dan dikelola AWS untuk Anda, di panel navigasi pilih Kunci terkelola AWS.

- Untuk melihat tombol di akun yang Anda buat dan kelola, di panel navigasi pilih Kunci terkelola pelanggan.

AWS KMS mendukung AWS CloudTrail, sehingga Anda dapat mengaudit penggunaan kunci untuk memverifikasi bahwa kunci digunakan dengan tepat. Kunci AWS KMS Anda dapat digunakan bersama AWS DMS dan layanan AWS yang didukung seperti Amazon RDS, Amazon S3, Amazon Redshift, dan Amazon EBS.

Anda juga dapat membuat kunci AWS KMS kustom untuk mengenkripsi data target untuk titik akhir AWS DMS berikut:

- Amazon Redshift – Untuk informasi selengkapnya, lihat [Membuat dan menggunakan kunci AWS KMS untuk mengenkripsi data target Amazon Redshift](#).
- Amazon S3 – Untuk informasi selengkapnya, lihat [Membuat kunci AWS KMS untuk mengenkripsi objek target Amazon S3](#).

Setelah Anda membuat AWS DMS sumber daya dengan kunci KMS, Anda tidak dapat mengubah kunci enkripsi untuk sumber daya tersebut. Pastikan Anda menentukan persyaratan kunci enkripsi sebelum membuat sumber daya AWS DMS Anda.

Keamanan jaringan untuk AWS Database Migration Service

Persyaratan keamanan untuk jaringan yang Anda buat saat menggunakan AWS Database Migration Service tergantung pada bagaimana Anda mengonfigurasi jaringan tersebut. Aturan umum untuk keamanan jaringan untuk AWS DMS adalah sebagai berikut:

- Instans replikasi harus memiliki akses ke titik akhir sumber dan target. Grup keamanan untuk instans replikasi harus memiliki ACL jaringan atau aturan yang mengizinkan keluar dari instans pada port basis data untuk titik akhir basis data.
- Titik akhir basis data harus mencakup ACL jaringan dan aturan grup keamanan yang mengizinkan akses masuk dari instans replikasi. Anda dapat mencapai hal ini menggunakan grup keamanan instans replikasi, alamat IP privat, alamat IP publik, atau alamat publik gateway NAT, tergantung pada konfigurasi Anda.
- Jika jaringan Anda menggunakan terowongan VPN, instans Amazon EC2 yang bertindak sebagai gateway NAT harus menggunakan grup keamanan yang memiliki aturan yang mengizinkan instans replikasi untuk mengirim lalu lintas melalui grup tersebut.

Secara default, grup keamanan VPC yang digunakan oleh instans replikasi AWS DMS memiliki aturan yang mengizinkan keluar ke 0.0.0.0/0 pada semua port. Jika Anda mengubah grup keamanan ini atau menggunakan grup keamanan Anda sendiri, keluar harus, minimal, diizinkan untuk titik akhir sumber dan target pada port basis data masing-masing.

Masing-masing konfigurasi jaringan yang dapat Anda gunakan untuk migrasi basis data memerlukan pertimbangan keamanan khusus:

- [Konfigurasi dengan semua komponen migrasi basis data dalam satu VPC](#) - Grup keamanan yang digunakan oleh titik akhir harus mengizinkan ingress pada port basis data dari instans replikasi. Pastikan bahwa grup keamanan yang digunakan oleh instans replikasi memiliki ingress ke titik akhir, atau Anda dapat membuat aturan dalam grup keamanan yang digunakan oleh titik akhir yang memungkinkan alamat IP privat akses instans replikasi.
- [Konfigurasi dengan beberapa VPC](#) – Grup keamanan yang digunakan oleh instans replikasi harus memiliki aturan untuk rentang VPC dan port DB pada basis data.
- [Konfigurasi untuk jaringan ke VPC menggunakan AWS Direct Connect atau VPN](#) – terowongan VPN memungkinkan lalu lintas ke terowongan dari VPC ke VPN on premise. Dalam konfigurasi ini, VPC mencakup aturan perutean yang mengirimkan lalu lintas yang ditujukan untuk alamat atau rentang IP tertentu ke host yang dapat menjembatani lalu lintas dari VPC ke VPN on premise.

Dalam kasus ini, host NAT mencakup pengaturan Grup Keamanan sendiri yang harus mengizinkan lalu lintas dari alamat IP privat Instans Replikasi atau grup keamanan ke instans NAT.

- [Konfigurasi untuk jaringan ke VPC menggunakan internet](#) – Grup keamanan VPC harus mencakup aturan perutean yang mengirim lalu lintas yang tidak ditujukan untuk VPC ke gateway Internet. Dalam konfigurasi ini, koneksi ke titik akhir tampaknya datang dari alamat IP publik pada instans replikasi.
- [Konfigurasi dengan instans RDS DB tidak dalam VPC ke instance DB di VPC menggunakan ClassicLink](#)— Ketika sumber atau target instans Amazon RDS DB tidak ada dalam VPC dan tidak berbagi grup keamanan dengan VPC tempat instance replikasi berada, Anda dapat mengatur server proxy dan ClassicLink menggunakannya untuk menghubungkan basis data sumber dan target.
- Titik akhir sumber berada di luar VPC yang digunakan oleh instans replikasi dan menggunakan gateway NAT – Anda dapat mengonfigurasi gateway penerjemahan alamat jaringan (NAT) menggunakan alamat IP Elastis tunggal yang terikat ke antarmuka jaringan Elastis tunggal. Antarmuka jaringan elastis ini kemudian menerima pengidentifikasi NAT (nat-#####). Jika VPC mencakup rute default ke gateway NAT tersebut alih-alih gateway internet, maka instans replikasi muncul untuk menghubungi titik akhir basis data menggunakan alamat IP publik gateway internet. Dalam kasus ini, ingress ke titik akhir basis data di luar VPC perlu mengizinkan ingress dari alamat NAT alih-alih alamat IP publik instans replikasi.
- VPC endpoint untuk mesin non-RDBMS – AWS DMS tidak mendukung VPC endpoint untuk mesin non-RDBMS.

Menggunakan SSL dengan AWS Database Migration Service

Anda dapat mengenkripsi koneksi untuk titik akhir sumber dan target menggunakan Lapisan Soket Aman (SSL). Untuk melakukannya, Anda dapat menggunakan Konsol Manajemen AWS DMS atau AWS DMS API untuk menetapkan sertifikat ke titik akhir. Anda juga dapat menggunakan konsol AWS DMS untuk mengelola sertifikat Anda.

Tidak semua basis data menggunakan SSL dengan cara yang sama. Amazon Aurora Edisi Kompatibel MySQL menggunakan nama server, titik akhir dari instans primer dalam kluster, sebagai titik akhir untuk SSL. Titik akhir Amazon Redshift sudah menggunakan koneksi SSL dan tidak memerlukan koneksi SSL yang diatur oleh AWS DMS. Titik akhir Oracle memerlukan langkah-langkah tambahan; untuk informasi selengkapnya, lihat [Dukungan SSL untuk titik akhir Oracle](#).

Topik

- [Keterbatasan menggunakan SSL dengan AWS DMS](#)
- [Mengelola sertifikat](#)
- [Mengaktifkan SSL untuk titik akhir yang kompatibel dengan MySQL, PostgreSQL, atau SQL Server](#)

Untuk menetapkan sertifikat ke titik akhir, Anda memberikan sertifikat root atau rantai sertifikat CA menengah yang mengarah ke root (sebagai paket sertifikat), yang digunakan untuk menandatangani sertifikat SSL server yang di-deploy pada titik akhir Anda. Sertifikat hanya diterima sebagai file X509 yang diformat PEM. Ketika Anda mengimpor sertifikat, Anda menerima Amazon Resource Name (ARN) yang dapat Anda gunakan untuk menentukan sertifikat untuk titik akhir. Jika Anda menggunakan Amazon RDS, Anda dapat mengunduh CA root dan paket sertifikat yang disediakan di file `rds-combined-ca-bundle.pem` yang dihosting oleh Amazon RDS. Untuk informasi selengkapnya tentang pengunduhan file ini, lihat [Menggunakan SSL/TLS untuk mengenkripsi sambungan ke instans DB](#) dalam Panduan Pengguna Amazon RDS.

Anda dapat memilih dari beberapa mode SSL yang akan digunakan untuk verifikasi sertifikat SSL Anda.

- `none` – Koneksi tidak dienkripsi. Opsi ini tidak aman, namun membutuhkan lebih sedikit overhead.
- `require` – Koneksi dienkripsi menggunakan SSL (TLS) tetapi verifikasi CA tidak dibuat. Pilihan ini lebih aman, dan membutuhkan lebih banyak overhead.
- `verify-ca` – Koneksi dienkripsi. Pilihan ini lebih aman, dan membutuhkan lebih banyak overhead. Opsi ini memverifikasi sertifikat server.

- **verify-full** – Koneksi dienkripsi. Pilihan ini lebih aman, dan membutuhkan lebih banyak overhead. Opsi ini memverifikasi sertifikat server dan memverifikasi bahwa nama host server cocok dengan atribut nama host untuk sertifikat tersebut.

Tidak semua mode SSL bekerja dengan semua titik akhir basis data. Tabel berikut menunjukkan mode SSL mana yang didukung untuk setiap mesin basis data.

Mesin DB	tidak ada	membutuhkan	verifikasi-ca	verifikasi-penuh
MySQL/MariaDB/Amazon Aurora MySQL	Default	Tidak didukung	Didukung	Didukung
Microsoft SQL Server	Default	Didukung	Tidak Didukung	Didukung
PostgreSQL	Default	Didukung	Didukung	Didukung
Amazon Redshift	Default	SSL tidak diaktifkan	SSL tidak diaktifkan	SSL tidak diaktifkan
Oracle	Default	Tidak didukung	Didukung	Tidak Didukung
SAP ASE	Default	SSL tidak diaktifkan	SSL tidak diaktifkan	Didukung
MongoDB	Default	Didukung	Tidak Didukung	Didukung
Db2 LUW	Default	Tidak Didukung	Didukung	Tidak Didukung
Db2 untuk z/OS	Default	Tidak Didukung	Didukung	Tidak Didukung

Note

Opsi Mode SSL pada konsol DMS atau API tidak berlaku untuk beberapa data streaming dan layanan NoSQL seperti Kinesis, dan DynamoDB. Secara default mereka aman, sehingga DMS menunjukkan bahwa pengaturan mode SSL sama dengan tidak ada (Mode SSL=None). Anda tidak perlu menyediakan konfigurasi tambahan untuk titik akhir Anda untuk menggunakan SSL. Misalnya, saat menggunakan Kinesis sebagai titik akhir target, Kinesis sudah aman secara default. Semua panggilan API ke Kinesis menggunakan SSL, sehingga

pilihan SSL tambahan di titik akhir DMS tidak diperlukan. Anda dapat memasukkan data dan mengambil data dengan aman melalui titik akhir SSL menggunakan protokol HTTPS, yang digunakan DMS secara default saat terhubung ke Kinesis Data Stream.

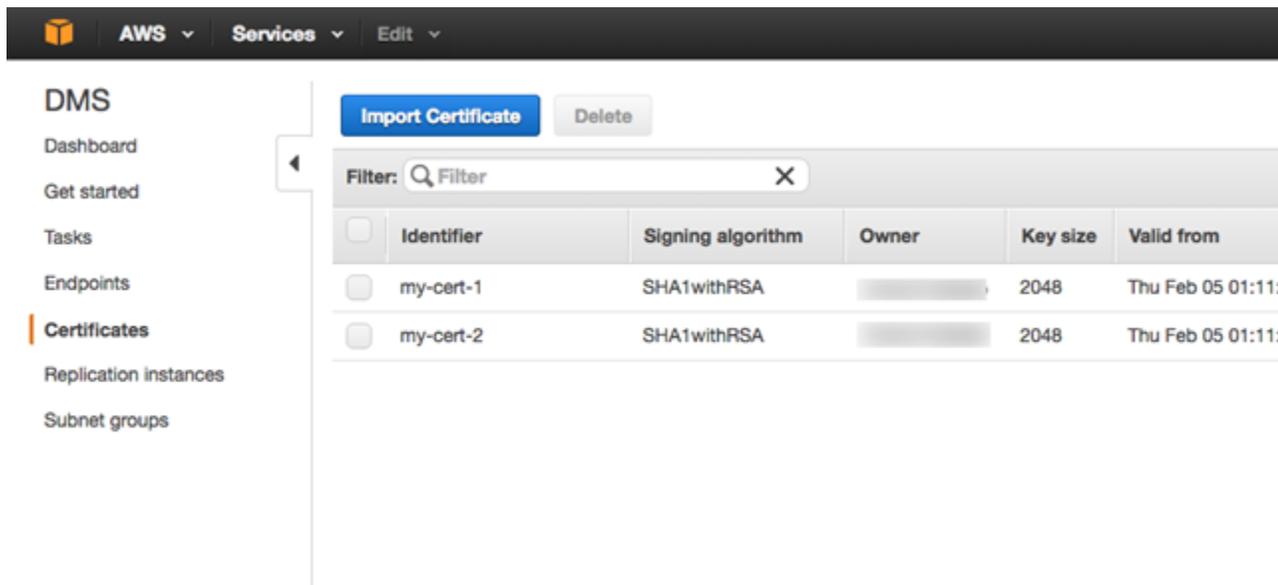
Keterbatasan menggunakan SSL dengan AWS DMS

Berikut ini adalah keterbatasan menggunakan SSL dengan AWS DMS:

- Koneksi SSL ke titik akhir target Amazon Redshift tidak didukung. AWS DMS menggunakan bucket Amazon S3 untuk mentransfer data ke basis data Amazon Redshift. Transmisi ini dienkripsi oleh Amazon Redshift secara default.
- Waktu habis SQL dapat terjadi saat melakukan tugas ubah tangkapan data (CDC) dengan titik akhir Oracle yang SSL-nya diaktifkan. Jika Anda memiliki masalah di mana penghitung CDC tidak mencerminkan angka yang diharapkan, atur parameter `MinimumTransactionSize` di bagian `ChangeProcessingTuning` dari pengaturan tugas ke nilai yang lebih rendah. Anda bisa mulai dengan nilai serendah 100. Untuk informasi tentang parameter `MinimumTransactionSize`, lihat [Mengubah pengaturan penyetelan pemrosesan](#).
- Anda dapat mengimpor sertifikat hanya dalam format `.pem` dan `.sso` (dompet Oracle).
- Dalam beberapa kasus, sertifikat SSL server Anda mungkin ditandatangani oleh otoritas sertifikat (CA) menengah. Jika demikian, pastikan bahwa seluruh rantai sertifikat yang mengarah dari CA menengah hingga CA root diimpor sebagai file `.pem` tunggal.
- Jika Anda menggunakan sertifikat yang ditandatangani sendiri pada server Anda, pilih `require` sebagai mode SSL Anda. Mode SSL `require` secara implisit mempercayai sertifikat SSL server dan tidak mencoba untuk memvalidasi apakah sertifikat telah ditandatangani oleh CA.

Mengelola sertifikat

Anda dapat menggunakan konsol DMS untuk melihat dan mengelola sertifikat SSL Anda. Anda juga dapat mengimpor sertifikat Anda menggunakan konsol DMS.



Mengaktifkan SSL untuk titik akhir yang kompatibel dengan MySQL, PostgreSQL, atau SQL Server

Anda dapat menambahkan koneksi SSL ke titik akhir yang baru dibuat atau titik akhir yang sudah ada.

Untuk membuat titik akhir AWS DMS dengan SSL

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.

Jika Anda masuk sebagai pengguna AWS Identity and Access Management (IAM), pastikan bahwa Anda memiliki izin yang sesuai untuk mengakses AWS DMS. Untuk informasi lebih lanjut tentang izin yang diperlukan untuk migrasi basis data, lihat [Izin IAM diperlukan untuk menggunakan AWS DMS](#).

2. Di panel navigasi, pilih Sertifikat.
3. Pilih Impor Sertifikat.
4. Unggah sertifikat yang ingin Anda gunakan untuk mengenkripsi koneksi ke titik akhir.

Note

Anda juga dapat mengunggah sertifikat menggunakan konsol AWS DMS saat Anda membuat atau mengubah titik akhir dengan memilih Tambahkan sertifikat CA baru pada halaman Buat titik akhir basis data.

Untuk Aurora Serverless sebagai target, dapatkan sertifikat yang disebutkan dalam [Menggunakan TLS/SSL dengan Aurora Serverless](#).

5. Buat titik akhir seperti yang dijelaskan dalam [Langkah 2: Tentukan titik akhir sumber dan target](#)

Untuk mengubah titik akhir AWS DMS yang sudah ada untuk menggunakan SSL

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.

Jika Anda masuk sebagai pengguna IAM, pastikan Anda memiliki izin yang sesuai untuk mengakses AWS DMS. Untuk informasi lebih lanjut tentang izin yang diperlukan untuk migrasi basis data, lihat [Izin IAM diperlukan untuk menggunakan AWS DMS](#).

2. Di panel navigasi, pilih Sertifikat.
3. Pilih Impor Sertifikat.
4. Unggah sertifikat yang ingin Anda gunakan untuk mengenkripsi koneksi ke titik akhir.

 Note

Anda juga dapat mengunggah sertifikat menggunakan konsol AWS DMS saat Anda membuat atau mengubah titik akhir dengan memilih Tambahkan sertifikat CA baru pada halaman Buat titik akhir basis data.

5. Di panel navigasi, pilih Titik akhir, pilih titik akhir yang ingin Anda ubah, dan pilih Modifikasi.
6. Pilih nilai untuk Mode SSL.

Jika Anda memilih mode verify-ca atau verify-full, tetapkan sertifikat yang ingin Anda gunakan untuk Sertifikat CA, seperti yang ditunjukkan berikut ini.

Create database endpoint

A database endpoint is used by the replication server to connect to a database. The database specified in the endpoint can be on-prem. Details should be specified in the form below. It is recommended that you test your endpoint connections here to avoid errors during pr

Endpoint type* Source Target ⓘ

Endpoint identifier* ⓘ

Source engine* ⓘ

Server name*

Port*

SSL mode* ⓘ

CA certificate* ⓘ

[Add new CA certificate](#)

User name*

Password*

▶ Advanced

7. Pilih Modifikasi.
8. Ketika titik akhir telah dimodifikasi, pilih titik akhir dan pilih Uji koneksi untuk menentukan apakah sambungan SSL berfungsi.

Setelah Anda membuat titik akhir sumber dan target, buat tugas yang menggunakan titik akhir ini. Untuk informasi lebih lanjut tentang cara membuat tugas, lihat [Langkah 3: Buat tugas dan migrasi data](#).

Pengubahan kata sandi basis data

Dalam sebagian besar situasi, mengubah kata sandi basis data untuk titik akhir sumber atau target Anda adalah hal yang mudah. Jika Anda perlu mengubah kata sandi basis data untuk titik akhir yang saat ini Anda gunakan dalam tugas migrasi atau replikasi, proses ini sedikit lebih kompleks. Prosedur berikut menunjukkan cara melakukannya.

Untuk mengubah kata sandi basis data untuk titik akhir dalam tugas migrasi atau replikasi

1. Masuk ke AWS Management Console dan buka konsol AWS DMS di <https://console.aws.amazon.com/dms/v2/>.

Jika Anda masuk sebagai pengguna IAM, pastikan Anda memiliki izin yang sesuai untuk mengakses AWS DMS. Untuk informasi lebih lanjut tentang izin yang diperlukan, lihat [Izin IAM diperlukan untuk menggunakan AWS DMS](#).

2. Di panel navigasi, pilih Tugas migrasi database.
3. Pilih tugas yang menggunakan titik akhir yang kata sandi basis datanya ingin Anda ubah, dan kemudian pilih Berhenti.
4. Saat tugas dihentikan, Anda dapat mengubah kata sandi basis data untuk titik akhir menggunakan alat asli yang Anda gunakan untuk bekerja dengan basis data.
5. Kembali ke Konsol Manajemen DMS dan pilih Titik akhir dari panel navigasi.
6. Pilih titik akhir untuk basis data yang kata sandinya Anda ubah, dan kemudian pilih Modifikasi.
7. Ketik kata sandi baru di kotak Kata Sandi, lalu pilih Simpan.
8. Pilih Tugas migrasi database dari panel navigasi.
9. Pilih tugas yang Anda hentikan sebelumnya, dan pilih Restart/Resume.
10. Pilih Mulai Ulang atau Lanjutkan, tergantung pada bagaimana Anda ingin melanjutkan tugas, lalu pilih Mulai tugas.

Kuota untuk AWS Database Migration Service

Setelah itu, Anda dapat menemukan kuota sumber daya dan kendala penamaan untuk AWS Database Migration Service (AWS DMS).

Pengukuran maksimum basis data yang AWS DMS dapat melakukan migrasi tergantung pada sejumlah faktor. Ini termasuk lingkungan sumber Anda, distribusi data dalam basis data sumber, dan seberapa sibuk sistem sumber Anda.

Cara terbaik untuk menentukan apakah sistem khusus Anda merupakan kandidat untuk AWS DMS adalah dengan mengujinya. Sebaiknya mulai perlahan sehingga Anda bisa mendapatkan konfigurasi berfungsi, kemudian menambahkan beberapa objek yang kompleks. Akhirnya, mencoba beban penuh sebagai pengujian.

Kuota sumber daya AWS Database Migration Service

Setiap akun AWS memiliki kuota untuk setiap Wilayah AWS, terkait jumlah sumber daya AWS DMS yang dapat dibuat. Setelah kuota sumber daya tercapai, panggilan tambahan untuk membuat sumber daya itu gagal dengan pengecualian.

Tabel berikut mencantumkan sumber daya AWS DMS dan kuota mereka tiap Wilayah AWS.

Sumber Daya	Kuota default
Throttling permintaan API	200 permintaan maksimum per detik
Tingkat penyegaran permintaan API	8 permintaan per detik
Contoh replikasi per akun pengguna	60
Total jumlah penyimpanan untuk instans replikasi	30,000 GB
Langganan acara per akun pengguna	60
Grup subnet replikasi per akun pengguna	60
Subnet per grup subnet replikasi	60

Sumber Daya	Kuota default
Titik akhir per akun pengguna	1.000
Titik akhir per instans replikasi	100
Tugas per akun pengguna	600
Tugas per instans replikasi	200
Sertifikat per akun pengguna	100
Penyedia data per akun pengguna	1.000
Profil instans per akun pengguna	60
Proyek migrasi per akun pengguna	10
Pengumpul data DMS per akun pengguna	10
Rekomendasi target yang dihasilkan pada satu waktu	100
Jumlah file yang dapat diunggah oleh pengumpul data DMS per jam	500
Migrasi data homogen per akun pengguna	600
Migrasi data homogen yang berjalan pada satu waktu	100
Migrasi data homogen per proyek migrasi	10
Replikasi tanpa server	100

Untuk informasi lebih lanjut tentang kuota throttling permintaan API dan tingkat penyegaran, lihat [Memahami throttling permintaan API](#).

Kuota 30.000-GB untuk penyimpanan berlaku untuk semua instans replikasi AWS DMS dalam Wilayah AWS tertentu. Penyimpanan ini digunakan untuk perubahan cache jika target tidak dapat menyesuaikan dengan sumber, dan untuk penyimpanan informasi yang mencatat perubahan itu.

Memahami throttling permintaan API

AWS DMS mendukung kuota permintaan API yang bervariasi, tetapi maksimum 200 panggilan API per detik. Dengan kata lain, permintaan API Anda dikenakan throttling saat melebihi tingkat ini. Selain itu, Anda dapat dibatasi hingga panggilan API yang lebih sedikit per detik, tergantung pada berapa lama waktu yang dibutuhkan AWS DMS untuk menyegarkan kembali kuota sebelum Anda membuat permintaan API lain. Kuota ini berlaku ketika Anda melakukan panggilan API secara langsung dan ketika panggilan API tersebut dilakukan atas nama Anda sebagai bagian dari penggunaan Konsol Manajemen AWS DMS.

Untuk memahami bagaimana throttling permintaan API bekerja, akan sangat membantu bila membayangkan bahwa AWS DMS memelihara bucket token yang melacak permintaan API Anda. Dalam skenario ini, setiap token dalam bucket mengizinkan Anda membuat satu panggilan API. Anda dapat memiliki tidak lebih dari 200 token dalam ember pada satu waktu. Saat Anda membuat panggilan API, AWS DMS menghapus satu token dari bucket. Jika Anda melakukan 200 panggilan API dalam waktu kurang dari satu detik, bucket Anda kosong dan setiap upaya untuk membuat panggilan API lain gagal. Untuk setiap detik Anda tidak melakukan panggilan API, AWS DMS tambahkan 8 token ke bucket, hingga maksimum 200 token. Ini adalah tingkat penyegaran permintaan AWS DMS API. Pada setiap titik setelah throttling, ketika Anda memiliki token yang ditambahkan ke bucket, Anda dapat membuat sebanyak mungkin panggilan API tambahan selama token tersedia sampai panggilan dikenakan throttling lagi.

Jika Anda menggunakan AWS CLI untuk menjalankan panggilan API yang dikenakan throttling, AWS DMS mengembalikan kesalahan seperti berikut:

```
An error occurred (ThrottlingException) when calling the AwsDmsApiCall operation
(reached max retries: 2): Rate exceeded
```

Di sini, *AwsDmsApiCall* adalah nama operasi API AWS DMS yang dikenakan throttling, misalnya, `DescribeTableStatistics`. Anda kemudian dapat mencoba lagi atau membuat panggilan yang berbeda setelah penundaan yang cukup untuk menghindari throttling.

Note

Tidak seperti throttling permintaan API yang dikelola oleh beberapa layanan lainnya, seperti Amazon EC2, Anda tidak dapat memerintahkan peningkatan kuota throttling permintaan API yang dikelola oleh AWS DMS.

Pemecahan masalah tugas migrasi di AWS Database Migration Service

Berikut ini, Anda dapat menemukan topik tentang pemecahan masalah dengan AWS Database Migration Service (AWS DMS). Topik ini dapat membantu Anda menyelesaikan masalah umum menggunakan AWS DMS dan basis data titik akhir yang dipilih.

Jika Anda telah membuka kasus AWS Support, teknisi dukungan Anda mungkin mengidentifikasi potensi masalah dengan salah satu konfigurasi basis data titik akhir Anda. Teknisi Anda mungkin juga meminta Anda untuk menjalankan skrip dukungan untuk mengembalikan informasi diagnostik tentang basis data Anda. Untuk detail tentang mengunduh, menjalankan, dan mengunggah informasi diagnostik dari skrip dukungan jenis ini, lihat [Bekerja dengan skrip dukungan diagnostik di AWS DMS](#).

Untuk tujuan pemecahan masalah, AWS DMS kumpulkan file trace dan dump di instance replikasi. Anda dapat memberikan file-file ini ke AWS Support jika terjadi masalah yang memerlukan pemecahan masalah. Secara default, DMS membersihkan file trace dan dump yang lebih tua dari tiga puluh hari. Untuk memilih keluar dari koleksi file trace dan dump, buka case dengan AWS Support.

Topik

- [Tugas migrasi berjalan perlahan](#)
- [Bilah status tugas tidak bergerak](#)
- [Tugas selesai tapi tidak ada yang dimigrasi](#)
- [Kunci asing dan indeks sekunder hilang](#)
- [AWS DMS tidak membuat CloudWatch log](#)
- [Masalah terjadi saat terhubung ke Amazon RDS](#)
- [Masalah jaringan terjadi](#)
- [CDC tertahan setelah beban penuh](#)
- [Kesalahan pelanggaran kunci primer terjadi saat Anda memulai ulang tugas](#)
- [Beban awal skema gagal](#)
- [Tugas gagal dengan kesalahan yang tidak diketahui](#)
- [Pemulaian ulang tugas memuat tabel dari awal](#)
- [Jumlah tabel per tugas menyebabkan masalah](#)

- [Tugas gagal ketika kunci primer dibuat pada kolom LOB](#)
- [Catatan duplikat terjadi pada tabel target tanpa kunci primer](#)
- [Titik akhir sumber termasuk dalam rentang IP yang dicadangkan](#)
- [Stempel waktu kacau dalam pertanyaan Amazon Athena](#)
- [Pemecahan masalah dengan Oracle](#)
- [Pemecahan masalah dengan MySQL](#)
- [Pemecahan masalah dengan PostgreSQL](#)
- [Pemecahan masalah Microsoft SQL Server](#)
- [Pemecahan masalah dengan Amazon Redshift](#)
- [Pemecahan masalah dengan Amazon Aurora MySQL](#)
- [Memecahkan masalah dengan SAP ASE](#)
- [Memecahkan masalah dengan IBM Db2](#)
- [Memecahkan masalah latensi di AWS Database Migration Service](#)
- [Bekerja dengan skrip dukungan diagnostik di AWS DMS](#)
- [Bekerja dengan dukungan AWS DMS diagnostik AMI](#)

Tugas migrasi berjalan perlahan

Beberapa masalah dapat menyebabkan tugas migrasi berjalan lambat, atau menyebabkan tugas berikutnya berjalan lebih lambat daripada tugas awal.

Alasan paling umum dari tugas migrasi yang berjalan lambat adalah adanya sumber daya tidak memadai yang dialokasikan untuk instans replikasi AWS DMS. Untuk memastikan bahwa instans Anda memiliki sumber daya yang cukup untuk tugas yang Anda jalankan, periksa penggunaan CPU, memori, swap file, dan IOPS instans replikasi Anda. Misalnya, beberapa tugas dengan Amazon Redshift sebagai titik akhir bersifat intensif I/O. Anda dapat meningkatkan IOPS untuk instans replikasi Anda atau membagi tugas Anda di beberapa instans replikasi untuk migrasi yang lebih efisien.

Untuk informasi lebih lanjut tentang penentuan ukuran instans replikasi Anda, lihat [Memilih ukuran terbaik untuk contoh replikasi](#).

Anda dapat meningkatkan kecepatan beban migrasi awal dengan cara melakukan hal berikut:

- Jika target Anda adalah instans DB Amazon RDS, pastikan bahwa Multi-AZ tidak diaktifkan untuk instans DB target.
- Matikan pencadangan otomatis atau pencatatan di basis data target selama pemuatan, dan aktifkan kembali fitur tersebut setelah migrasi selesai.
- Jika fitur tersedia pada target Anda, gunakan provisioned IOPS.
- Jika data migrasi Anda berisi LOB, pastikan bahwa tugas dioptimalkan untuk migrasi LOB. Untuk informasi lebih lanjut tentang mengoptimalkan untuk LOB, lihat [Menargetkan pengaturan tugas metadata](#).

Bilah status tugas tidak bergerak

Bilah status tugas memberikan estimasi tentang kemajuan tugas. Kualitas estimasi ini tergantung pada kualitas statistik tabel basis data sumber; semakin baik statistik tabel, semakin akurat estimasi.

Untuk tugas dengan hanya satu tabel yang tidak memiliki estimasi statistik baris, AWS DMS tidak dapat memberikan persentase estimasi lengkap apa pun. Dalam kasus ini, gunakan status tugas dan indikasi baris yang dimuat untuk mengonfirmasi bahwa tugas berjalan dan membuat kemajuan.

Tugas selesai tapi tidak ada yang dimigrasi

Lakukan hal berikut jika tidak ada yang dimigrasi setelah tugas Anda selesai.

- Periksa apakah pengguna yang membuat titik akhir telah membaca akses ke tabel yang ingin Anda migrasi.
- Periksa apakah objek yang ingin Anda migrasi adalah tabel. Jika objek tersebut adalah tampilan, perbarui pemetaan tabel dan tentukan pencari objek sebagai “tampilan” atau “semua”. Untuk informasi selengkapnya, lihat [Menentukan pemilihan tabel dan transformasi aturan dari konsol](#).

Kunci asing dan indeks sekunder hilang

AWS DMS membuat tabel, kunci primer, dan dalam beberapa kasus indeks unik, tetapi tidak membuat objek lain yang tidak diperlukan untuk memigrasi data dari sumber secara efisien. Misalnya, DMS tidak membuat indeks sekunder, kendala kunci non-primer, atau default data.

Untuk memigrasi objek sekunder dari basis data Anda, gunakan alat asli basis data jika Anda bermigrasi ke mesin basis data yang sama dengan basis data sumber Anda. Gunakan AWS Schema

Conversion Tool (AWS SCT) jika Anda bermigrasi ke mesin basis data yang berbeda dari yang digunakan oleh basis data sumber Anda untuk memigrasi objek sekunder.

AWS DMStidak membuat CloudWatch log

Jika tugas replikasi Anda tidak membuat CloudWatch log, pastikan akun Anda memiliki `dms-cloudwatch-logs-role` peran tersebut. Jika peran ini tidak ada, lakukan hal berikut untuk membuatnya:

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Pilih tab Peran. Pilih Buat peran.
3. Di bagian Pilih jenis entitas tepercaya, pilih Layanan AWS.
4. Di bagian Pilih kasus penggunaan, pilih DMS.
5. Pilih Selanjutnya: Izin.
6. Masukkan **AmazonDMSCloudWatchLogsRole** di bidang pencarian, dan centang kotak di sebelah CloudWatchLogsRoleAmazonDMS. Ini memberikan AWS DMS izin untuk mengakses CloudWatch
7. Pilih Berikutnya: Tanda.
8. Pilih Berikutnya: Tinjauan.
9. Masukkan **dms-cloudwatch-logs-role** untuk nama Peran. Nama ini peka huruf besar/kecil.
10. Pilih Buat peran.

Masalah terjadi saat terhubung ke Amazon RDS

Ada beberapa alasan mengapa Anda tidak dapat terhubung ke instans DB Amazon RDS yang Anda tetapkan sebagai sumber atau target. Berikut beberapa item untuk diperiksa:

- Periksa apakah kombinasi nama pengguna dan kata sandi sudah benar.
- Periksa apakah nilai titik akhir yang ditampilkan di konsol Amazon RDS untuk instans tersebut sama dengan pengidentifikasi titik akhir yang Anda gunakan untuk membuat titik akhir AWS DMS.
- Periksa apakah nilai port yang ditampilkan di konsol Amazon RDS untuk instans tersebut sama dengan port yang ditetapkan untuk titik akhir AWS DMS.

- Periksa apakah grup keamanan yang ditetapkan ke instans DB Amazon RDS mengizinkan koneksi dari instans replikasi AWS DMS.
- Jika instans replikasi AWS DMS dan instans DB Amazon RDS tidak ada di dalam virtual private cloud (VPC) yang sama, periksa apakah instans DB dapat diakses publik.

Pesan kesalahan: koneksi string thread salah: nilai thread salah 0

Kesalahan ini dapat sering terjadi ketika Anda menguji koneksi ke titik akhir. Kesalahan ini menunjukkan bahwa ada kesalahan dalam string koneksi. Contohnya adalah spasi setelah alamat IP host. Contoh lain adalah karakter buruk yang disalin ke string koneksi.

Masalah jaringan terjadi

Masalah jaringan yang paling umum melibatkan grup keamanan VPC yang digunakan oleh instans replikasi AWS DMS. Secara default, grup keamanan ini memiliki aturan yang mengizinkan keluar ke 0.0.0.0/0 pada semua port. Dalam banyak kasus, Anda mengubah grup keamanan ini atau menggunakan grup keamanan Anda sendiri. Jika demikian, minimal, pastikan untuk memberikan jalan keluar ke titik akhir sumber dan target pada port basis data masing-masing.

Masalah terkait konfigurasi lainnya dapat mencakup hal berikut:

- Instans replikasi serta titik akhir sumber dan target di VPC yang sama – Grup keamanan yang digunakan oleh titik akhir harus mengizinkan ingress pada port basis data dari instans replikasi. Pastikan bahwa grup keamanan yang digunakan oleh instans replikasi memiliki ingress ke titik akhir. Atau Anda dapat membuat aturan dalam grup keamanan yang digunakan oleh titik akhir yang mengizinkan alamat IP privat dari akses instans replikasi.
- Titik akhir sumber berada di luar VPC yang digunakan oleh instans replikasi (menggunakan gateway internet) – Grup keamanan VPC harus menyertakan aturan perutean yang mengirim lalu lintas yang bukan ditujukan untuk VPC ke gateway internet. Dalam konfigurasi ini, koneksi ke titik akhir tampaknya datang dari alamat IP publik pada instans replikasi.
- Titik akhir sumber berada di luar VPC yang digunakan oleh instans replikasi (menggunakan gateway NAT) – Anda dapat mengonfigurasi gateway penerjemahan alamat jaringan (NAT) menggunakan alamat IP elastis tunggal yang terikat pada antarmuka jaringan elastis tunggal. Gateway NAT ini menerima pengidentifikasi NAT (nat-#####).

Dalam beberapa kasus, VPC mencakup rute default untuk gateway NAT alih-alih gateway internet. Dalam kasus seperti itu, instans replikasi muncul untuk menghubungi titik akhir basis data

menggunakan alamat IP publik gateway NAT. Di sini, ingress ke titik akhir basis data di luar VPC perlu mengizinkan ingress dari alamat NAT alih-alih alamat IP publik instans replikasi.

Untuk informasi tentang menggunakan server nama on premise Anda sendiri, lihat [Menggunakan server nama on-premise Anda sendiri](#).

CDC tertahan setelah beban penuh

Perubahan replikasi yang lambat atau tertahan dapat terjadi setelah migrasi beban penuh ketika beberapa pengaturan AWS DMS bertentangan satu sama lain.

Sebagai contoh, anggaplah bahwa parameter Mode persiapan tabel target diatur ke Tidak melakukan apa pun atau Memotong. Dalam hal ini, Anda telah menginstruksikan AWS DMS untuk tidak melakukan pengaturan pada tabel target, termasuk membuat indeks primer dan unik. Jika Anda belum membuat kunci primer atau unik pada tabel target, AWS DMS melakukan pemindaian tabel secara lengkap untuk setiap pembaruan. Pendekatan ini dapat memengaruhi performa secara signifikan.

Kesalahan pelanggaran kunci primer terjadi saat Anda memulai ulang tugas

Kesalahan ini dapat terjadi ketika data tetap berada dalam basis data target dari tugas migrasi sebelumnya. Jika opsi Mode persiapan tabel target diatur ke Tidak melakukan apa pun, AWS DMS tidak melakukan persiapan apa pun pada tabel target, termasuk membersihkan data yang dimasukkan dari tugas sebelumnya.

Untuk memulai ulang tugas Anda dan menghindari kesalahan ini, hapus baris yang dimasukkan ke dalam tabel target dari tugas yang dijalankan sebelumnya.

Beban awal skema gagal

Dalam beberapa kasus, beban awal skema Anda mungkin gagal dengan kesalahan `Operation:getSchemaListDetails:errType=, status=0, errMessage=, errDetails=.`

Dalam kasus tersebut, akun pengguna yang digunakan oleh AWS DMS untuk menyambung ke titik akhir sumber tidak memiliki izin yang diperlukan.

Tugas gagal dengan kesalahan yang tidak diketahui

Penyebab jenis kesalahan yang tidak diketahui dapat bervariasi. Namun, sering kali kita menemukan bahwa masalah ini melibatkan sumber daya yang tidak mencukupi yang dialokasikan ke instans replikasi AWS DMS.

Untuk memastikan bahwa instans replikasi Anda memiliki cukup sumber daya untuk melakukan migrasi, periksa penggunaan CPU, memori, swap file, dan IOPS instans Anda. Untuk informasi lebih lanjut tentang pemantauan, lihat [Metrik AWS Database Migration Service](#).

Pemulaian ulang tugas memuat tabel dari awal

AWS DMS memulai ulang tabel dan memuat dari awal ketika belum menyelesaikan beban awal tabel. Ketika tugas dimulai ulang, AWS DMS memuat ulang tabel dari awal ketika beban awal tidak selesai.

Jumlah tabel per tugas menyebabkan masalah

Tidak ada batas yang ditetapkan pada jumlah tabel per tugas replikasi. Namun, kami sarankan untuk membatasi jumlah tabel dalam tugas menjadi kurang dari 60.000, sebagai aturan praktis. Penggunaan sumber daya sering dapat menjadi hambatan ketika satu tugas menggunakan lebih dari 60.000 tabel.

Tugas gagal ketika kunci primer dibuat pada kolom LOB

Dalam mode LOB PENUH atau LOB TERBATAS, AWS DMS tidak mendukung replikasi kunci primer yang berjenis data LOB.

DMS awalnya memigrasi baris dengan kolom LOB sebagai nol, kemudian memperbarui kolom LOB. Jadi, ketika kunci primer dibuat pada kolom LOB, penyisipan awal gagal karena kunci primer tidak boleh nol. Sebagai solusi, tambahkan kolom lain sebagai kunci primer dan hapus kunci primer dari kolom LOB.

Catatan duplikat terjadi pada tabel target tanpa kunci primer

Menjalankan tugas beban penuh dan CDC dapat membuat catatan duplikat pada tabel target yang tidak memiliki kunci primer atau indeks unik. Untuk menghindari duplikasi catatan pada tabel target

selama tugas beban penuh dan CDC, pastikan bahwa tabel target memiliki kunci primer atau indeks unik.

Titik akhir sumber termasuk dalam rentang IP yang dicadangkan

Jika basis data sumber AWS DMS menggunakan alamat IP dalam rentang IP yang dicadangkan yaitu 192.168.0.0/24, pengujian koneksi titik akhir sumber gagal. Langkah-langkah berikut memberikan kemungkinan solusi:

1. Temukan satu instans Amazon EC2 yang tidak ada dalam rentang yang dicadangkan yang dapat berkomunikasi ke basis data sumber di 192.168.0.0/24.
2. Instal proksi socat dan jalankan. Bagian berikut menunjukkan satu contoh.

```
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
&
```

Gunakan alamat IP instans Amazon EC2 dan port database yang diberikan sebelumnya untuk titik akhir. AWS DMS Pastikan bahwa endpoint memiliki grup keamanan yang memungkinkan AWS DMS untuk mengakses port database. Perhatikan bahwa proxy harus berjalan selama durasi eksekusi tugas DMS Anda. Tergantung pada kasus penggunaan, Anda mungkin perlu mengotomatiskan pengaturan proxy.

Stempel waktu kacau dalam pertanyaan Amazon Athena

Jika stempel waktu kacau dalam kueri Athena, gunakan AWS Management Console atau [ModifyEndpoint](#) tindakan untuk menetapkan nilai titik akhir Amazon S3 `parquetTimestampInMillisecond` Anda. `true` Untuk informasi selengkapnya, lihat [S3Settings](#).

Pemecahan masalah dengan Oracle

Berikut ini, Anda dapat mempelajari tentang pemecahan masalah yang spesifik pada penggunaan AWS DMS dengan basis data Oracle.

Topik

- [Menarik data dari tampilan](#)
- [Memigrasi LOB dari Oracle 12c](#)
- [Beralih antara Oracle LogMiner dan Binary Reader](#)
- [Kesalahan: Oracle CDC berhenti 122301 penghitung coba kembali oracle CDC terlampaui.](#)
- [Secara otomatis menambahkan supplemental logging untuk titik akhir sumber Oracle](#)
- [Perubahan LOB tidak sedang ditangkap](#)
- [Kesalahan: ORA-12899: Nilai terlalu besar untuk kolom nama-kolom](#)
- [Tipe data ANGKA yang disalahartikan](#)
- [Catatan hilang selama beban penuh](#)
- [Kesalahan Tabel](#)
- [Kesalahan: Tidak dapat mengambil id tujuan log Redo yang diarsipkan Oracle](#)
- [Mengevaluasi kinerja baca Oracle redo atau arsip log](#)

Menarik data dari tampilan

Anda dapat menarik data satu kali dari tampilan; Anda tidak dapat menggunakannya untuk replikasi yang sedang berlangsung. Untuk dapat mengekstrak data dari tampilan, Anda harus menambahkan kode berikut ke bagian pengaturan Endpoint dari halaman endpoint sumber Oracle. Ketika Anda mengekstraksi data dari tampilan, tampilan ditampilkan sebagai tabel pada skema target.

```
"ExposeViews": true
```

Memigrasi LOB dari Oracle 12c

AWS DMS dapat menggunakan dua metode untuk menangkap perubahan pada database Oracle, Binary Reader dan LogMiner Oracle. Secara default, AWS DMS menggunakan Oracle LogMiner untuk menangkap perubahan. Namun, pada Oracle 12c, Oracle LogMiner tidak mendukung kolom LOB. Untuk menangkap perubahan kolom LOB pada Oracle 12c, gunakan Binary Reader.

Beralih antara Oracle LogMiner dan Binary Reader

AWS DMS dapat menggunakan dua metode untuk menangkap perubahan ke database Oracle sumber, Binary Reader dan LogMiner Oracle. Oracle LogMiner adalah default. Untuk beralih menggunakan Binary Reader untuk menangkap perubahan, lakukan hal berikut:

Untuk menggunakan binary reader untuk menangkap perubahan

1. Masuk ke AWS Management Console dan buka konsol AWS DMS tersebut di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih Titik akhir.
3. Pilih titik akhir sumber Oracle di mana Anda ingin menggunakan Binary Reader.
4. Pilih Modifikasi.
5. Pilih Lanjutan, dan kemudian tambahkan kode berikut untuk Atribut koneksi tambahan.

```
useLogminerReader=N
```

6. Gunakan alat developer Oracle seperti SQL-Plus untuk memberikan hak tambahan berikut kepada akun pengguna AWS DMS yang digunakan untuk terhubung ke titik akhir Oracle.

```
SELECT ON V_$TRANSPORTABLE_PLATFORM
```

Kesalahan: Oracle CDC berhenti 122301 penghitung coba kembali oracle CDC terlampaui.

Kesalahan ini terjadi ketika log arsip Oracle yang diperlukan telah dihapus dari server Anda sebelum AWS DMS dapat menggunakannya untuk menangkap perubahan. Tingkatkan kebijakan retensi log Anda di server basis data Anda. Untuk basis data Amazon RDS, jalankan prosedur berikut untuk meningkatkan retensi log. Sebagai contoh, kode berikut meningkatkan retensi log pada instans DB Amazon RDS hingga 24 jam.

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24);
```

Secara otomatis menambahkan supplemental logging untuk titik akhir sumber Oracle

Secara default, AWS DMS menonaktifkan supplemental logging. Untuk secara otomatis mengaktifkan supplemental logging untuk titik akhir sumber Oracle, lakukan hal berikut:

Untuk menambahkan supplemental logging untuk titik akhir sumber Oracle

1. Masuk ke AWS Management Console dan buka konsol AWS DMS tersebut di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih Titik akhir.
3. Pilih titik akhir sumber Oracle yang ingin Anda tambahkan supplemental logging.
4. Pilih Modifikasi.
5. Pilih Lanjutan, dan kemudian tambahkan kode berikut ke kotak teks Atribut koneksi tambahan:

```
addSupplementalLogging=Y
```

6. Pilih Modifikasi.

Perubahan LOB tidak sedang ditangkap

Saat ini, tabel harus memiliki kunci primer agar AWS DMS menangkap perubahan LOB. Jika tabel yang berisi LOB tidak memiliki kunci primer, ada beberapa tindakan yang dapat Anda lakukan untuk menangkap perubahan LOB:

- Tambahkan kunci primer ke tabel. Hal ini bisa sesederhana menambahkan kolom ID dan mengisinya dengan urutan menggunakan pemicu.
- Buat tampilan terwujud dari tabel yang mencakup ID yang dihasilkan sistem sebagai kunci primer dan migrasi tampilan terwujud alih-alih tabel.
- Buat siaga logis, tambahkan kunci primer ke tabel, dan bermigrasi dari siaga logis.

Kesalahan: ORA-12899: Nilai terlalu besar untuk kolom *nama-kolom*

Kesalahan "ORA-12899: nilai terlalu besar untuk kolom *nama-kolom*" sering disebabkan oleh beberapa masalah.

Dalam salah satu masalah ini, ada ketidakcocokan dalam set karakter yang digunakan oleh basis data sumber dan target.

Di masalah yang lain, pengaturan national language support (NLS) berbeda antara dua basis data. Penyebab umum dari kesalahan ini adalah ketika parameter NLS_LENGTH_SEMANTICS basis data sumber diatur ke CHAR dan parameter NLS_LENGTH_SEMANTICS basis data target diatur ke BYTE.

Tipe data ANGKA yang disalahartikan

Tipe data ANGKA Oracle diubah menjadi berbagai jenis data AWS DMS, tergantung pada presisi dan skala ANGKA. Konversi ini didokumentasikan di sini [Jenis data sumber untuk Oracle](#). Cara jenis NUMBER dikonversi juga dapat dipengaruhi dengan menggunakan pengaturan endpoint untuk sumber Oracle endpoint. Pengaturan titik akhir ini didokumentasikan dalam [Pengaturan titik akhir saat menggunakan Oracle sebagai sumber AWS DMS](#).

Catatan hilang selama beban penuh

Saat melakukan beban penuh, AWS DMS mencari transaksi terbuka pada tingkat basis data dan menunggu transaksi dilakukan. Sebagai contoh, berdasarkan pengaturan tugas `TransactionConsistencyTimeout=600`, AWS DMS menunggu selama 10 menit bahkan jika transaksi terbuka ada di tabel yang tidak termasuk dalam pemetaan tabel. Tetapi jika transaksi terbuka ada di tabel yang disertakan dalam pemetaan tabel, dan transaksi tidak dilakukan tepat waktu, catatan yang hilang dalam hasil tabel target.

Anda dapat mengubah pengaturan tugas `TransactionConsistencyTimeout` dan meningkatkan waktu tunggu jika Anda tahu bahwa transaksi terbuka akan memakan waktu lebih lama untuk dilakukan.

Selain itu, perhatikan bahwa nilai default pengaturan tugas `FailOnTransactionConsistencyBreached` adalah `false`. Ini berarti AWS DMS terus menerapkan transaksi lain tetapi transaksi terbuka terlewatkan. Jika Anda ingin tugas menjadi gagal ketika transaksi terbuka tidak ditutup tepat waktu, Anda dapat mengatur `FailOnTransactionConsistencyBreached` ke `true`.

Kesalahan Tabel

Table Error muncul dalam statistik tabel selama replikasi jika WHERE klausa tidak mereferensikan kolom kunci utama, dan logging tambahan tidak digunakan untuk semua kolom.

Untuk memperbaiki masalah ini, aktifkan pencatatan tambahan untuk semua kolom tabel yang direferensikan. Untuk informasi selengkapnya, lihat [Mengatur supplemental logging](#).

Kesalahan: Tidak dapat mengambil id tujuan log Redo yang diarsipkan Oracle

Kesalahan ini terjadi ketika sumber Oracle Anda tidak memiliki log arsip yang dihasilkan atau V \$ARCHIVED_LOG kosong. Anda dapat mengatasi kesalahan dengan mengganti log secara manual.

Untuk database Amazon RDS, jalankan prosedur berikut untuk mengganti file log.
switch_logfileProsedur ini tidak memiliki parameter.

```
exec rdsadmin.rdsadmin_util.switch_logfile;
```

Untuk database sumber Oracle yang dikelola sendiri, gunakan perintah berikut untuk memaksa sakelar log.

```
ALTER SYSTEM SWITCH LOGFILE ;
```

Mengevaluasi kinerja baca Oracle redo atau arsip log

Jika Anda mengalami masalah kinerja dengan sumber Oracle Anda, Anda dapat mengevaluasi kinerja baca redo Oracle atau log arsip Anda untuk menemukan cara untuk meningkatkan kinerja. Untuk menguji kinerja pembacaan log ulang atau arsip, gunakan [image mesin Amazon AWS DMS diagnostik](#) (AMI).

Anda dapat menggunakan AMI AWS DMS diagnostik untuk melakukan hal berikut:

- Gunakan metode BFile untuk mengevaluasi kinerja file redo log.
- Gunakan LogMiner metode ini untuk mengevaluasi kinerja file redo log.
- Gunakan metode PL/SQL (dbms_lob.read) untuk mengevaluasi kinerja file redo log.
- Gunakan Single-thread untuk mengevaluasi kinerja baca di AsmFile.

- Gunakan Multi-utas untuk mengevaluasi kinerja baca di AsmFile.
- Gunakan Direct OS Readfile () fungsi Windows atau Pread64 Linux untuk mengevaluasi file redo log.

Kemudian Anda dapat mengambil langkah-langkah perbaikan berdasarkan hasil.

Untuk menguji kinerja baca pada file log oracle redo atau arsip

1. Buat instans AMI Amazon EC2 AWS DMS diagnostik dan sambungkan ke instans tersebut.

Untuk informasi lebih lanjut lihat, [Bekerja dengan AMI AWS DMS diagnostik](#).

2. Jalankan perintah `awsreplperf`.

```
$ awsreplperf
```

Perintah menampilkan opsi AWS DMS Oracle Read Performance Utility.

```
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. Pilih opsi dari daftar.
4. Masukkan koneksi database berikut dan informasi log arsip.

```
Oracle user name [system]:
Oracle password:

Oracle connection name [orcl1x]:
Connection format hostname:port/instance

Oracle event trace? [N]:
Default N = No or Y = Yes

Path to redo or archive log file []:
```

- Periksa output yang ditampilkan untuk informasi kinerja baca yang relevan. Misalnya, berikut ini menunjukkan output yang dapat dihasilkan dari memilih opsi nomor 2, Baca menggunakan LogMiner.

```

Enter your choice>>2
Oracle user name: [system] >> * * *
Oracle password :
Oracle connection name : [orcl1x] >> * * * 0:1521/porcl
Oracle event trace ? : [N] >>n
Full path to redo or archive log file: [ ] >>+EBSFRA/PORCL/ONLINELOG/group_11.1380.1101828345
Elapsed Time : 7044.83973 sec
Read speed in : 0.088575 MB/sec
LogMinerRead: counted 1198389 redo log rows, total undo / redo size : 655073562
  
```

- Untuk keluar dari utilitas, masukkan 0 (nol).

Langkah selanjutnya

- Ketika hasil menunjukkan bahwa kecepatan baca di bawah ambang batas yang dapat diterima, jalankan [skrip dukungan diagnostik Oracle](#) di titik akhir, tinjau bagian Waktu Tunggu, Profil Muat, dan Profil IO. Kemudian sesuaikan konfigurasi abnormal apa pun yang mungkin meningkatkan kinerja baca. Misalnya, jika file log redo Anda hingga 2 GB, coba tingkatkan LOG_BUFFER menjadi 200 MB untuk membantu meningkatkan kinerja.
- Tinjau [Praktik AWS DMS Terbaik](#) untuk memastikan instans replikasi DMS, tugas, dan titik akhir dikonfigurasi secara optimal.

Pemecahan masalah dengan MySQL

Berikut ini, Anda dapat mempelajari tentang pemecahan masalah yang spesifik pada penggunaan AWS DMS dengan basis data MySQL.

Topik

- [Tugas CDC gagal untuk titik akhir instans DB Amazon RDS karena log biner dinonaktifkan](#)
- [Koneksi ke instans MySQL target terputus selama pengerjaan tugas](#)
- [Menambahkan autocommit ke titik akhir yang kompatibel dengan MySQL](#)
- [Menonaktifkan kunci asing pada titik akhir yang kompatibel dengan MySQL target](#)
- [Karakter diganti dengan tanda tanya](#)
- [Entri log "Peristiwa buruk"](#)

- [Mengubah tangkapan data dengan MySQL 5.5](#)
- [Meningkatkan retensi log biner untuk instans DB Amazon RDS](#)
- [Pesan log: Beberapa perubahan dari basis data sumber tidak memiliki dampak ketika diterapkan ke basis data target.](#)
- [Kesalahan: Pengidentifikasi terlalu panjang](#)
- [Kesalahan: Set karakter yang tidak didukung menyebabkan konversi data bidang gagal](#)
- [Kesalahan: Codepage 1252 ke UTF8 \[120112\] konversi data bidang gagal](#)
- [Indeks, Kunci Asing, atau Pembaruan atau Penghapusan Cascade Tidak Dimigrasi](#)

Tugas CDC gagal untuk titik akhir instans DB Amazon RDS karena log biner dinonaktifkan

Masalah ini terjadi dengan instans DB Amazon RDS karena pencadangan otomatis dinonaktifkan. Aktifkan pencadangan otomatis dengan mengatur periode retensi cadangan ke nilai selain nol.

Koneksi ke instans MySQL target terputus selama pengerjaan tugas

Jika Anda memiliki tugas dengan LOB yang terputus dari target MySQL, Anda mungkin melihat jenis kesalahan berikut dalam log tugas.

```
[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.
```

Dalam hal ini, Anda mungkin perlu menyesuaikan beberapa pengaturan tugas Anda.

Untuk mengatasi masalah di mana tugas terputus dari target MySQL, lakukan hal berikut:

- Periksa apakah Anda memiliki serangkaian variabel basis data `max_allowed_packet` yang cukup besar untuk menahan LOB terbesar Anda.
- Periksa apakah Anda memiliki serangkaian variabel berikut agar bisa memiliki nilai waktu habis yang besar. Kami sarankan agar Anda menggunakan nilai minimal 5 menit untuk masing-masing variabel ini.

- `net_read_timeout`
- `net_write_timeout`
- `wait_timeout`

[Untuk informasi tentang pengaturan variabel sistem MySQL, lihat Variabel Sistem Server dalam dokumentasi MySQL.](#)

Menambahkan autocommit ke titik akhir yang kompatibel dengan MySQL

Untuk menambahkan autocommit ke titik akhir yang kompatibel dengan MySQL

1. Masuk ke AWS Management Console dan buka konsol AWS DMS tersebut di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih Titik akhir.
3. Pilih titik akhir target yang kompatibel dengan MySQL yang ingin Anda tambahkan autocommit.
4. Pilih Modifikasi.
5. Pilih Lanjutan, dan kemudian tambahkan kode berikut ke kotak teks Atribut koneksi tambahan:

```
Initstmt= SET AUTOCOMMIT=1
```

6. Pilih Modifikasi.

Menonaktifkan kunci asing pada titik akhir yang kompatibel dengan MySQL target

Anda dapat menonaktifkan pemeriksaan kunci asing di MySQL dengan menambahkan hal berikut ke Atribut Koneksi Tambahan dalam bagian Lanjutan dari titik akhir MySQL, Amazon Aurora Edisi Kompatibel MySQL, atau MariaDB target.

Untuk menonaktifkan kunci asing pada titik akhir yang kompatibel dengan MySQL target

1. Masuk ke AWS Management Console dan buka konsol AWS DMS tersebut di <https://console.aws.amazon.com/dms/v2/>.
2. Pilih Titik akhir.

3. Pilih titik akhir MySQL, Aurora MySQL, atau MariaDB target yang kunci asingnya ingin Anda nonaktifkan.
4. Pilih Modifikasi.
5. Pilih Lanjutan, dan kemudian tambahkan kode berikut ke kotak teks Atribut koneksi tambahan:

```
Initstmt=SET FOREIGN_KEY_CHECKS=0
```

6. Pilih Modifikasi.

Karakter diganti dengan tanda tanya

Situasi yang paling umum yang menyebabkan masalah ini adalah ketika karakter titik akhir sumber telah dikodekan oleh set karakter yang tidak didukung AWS DMS.

Entri log "Peristiwa buruk"

Entri "peristiwa buruk" di log migrasi biasanya menunjukkan bahwa operasi bahasa definisi data (DDL) yang tidak didukung dicoba pada titik akhir basis data sumber. Operasi DDL yang tidak didukung menyebabkan peristiwa yang tidak dapat dilewati instans replikasi, sehingga peristiwa buruk dicatat.

Untuk mengatasi masalah ini, mulai ulang tugas dari awal. Melakukan hal ini memuat ulang tabel dan mulai menangkap perubahan pada titik setelah operasi DDL yang tidak didukung dikeluarkan.

Mengubah tangkapan data dengan MySQL 5.5

AWS DMS change data capture (CDC) untuk basis data Amazon RDS yang kompatibel dengan MySQL memerlukan log biner berbasis baris citra penuh, yang tidak didukung di MySQL versi 5.5 atau lebih rendah. Untuk menggunakan AWS DMS CDC, Anda harus meningkatkan instans DB Amazon RDS Anda ke MySQL versi 5.6.

Meningkatkan retensi log biner untuk instans DB Amazon RDS

AWS DMS memerlukan retensi berkas log biner untuk change data capture. Untuk meningkatkan retensi log pada instans DB Amazon RDS, jalankan prosedur berikut. Contoh berikut meningkatkan retensi log biner hingga 24 jam.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Pesan log: Beberapa perubahan dari basis data sumber tidak memiliki dampak ketika diterapkan ke basis data target.

Saat AWS DMS memperbarui nilai kolom basis data MySQL ke nilai yang ada, pesan `zero rows affected` dikembalikan dari MySQL. Perilaku ini tidak seperti mesin basis data lain seperti Oracle dan SQL Server. Mesin ini memperbarui satu baris, bahkan ketika nilai yang menggantikan sama dengan yang sekarang.

Kesalahan: Pengidentifikasi terlalu panjang

Kesalahan berikut terjadi saat pengidentifikasi terlalu panjang:

```
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)
```

Dalam beberapa kasus, Anda mengatur AWS DMS untuk membuat tabel dan kunci primer dalam basis data target. Dalam kasus ini, DMS saat ini tidak menggunakan nama yang sama untuk kunci primer yang digunakan dalam basis data sumber. Sebagai gantinya, DMS menciptakan nama kunci primer berdasarkan nama tabel. Ketika nama tabel panjang, pengidentifikasi yang dibuat secara otomatis dapat lebih panjang dari batas yang diizinkan untuk MySQL.

Untuk mengatasi masalah ini, pendekatan saat ini adalah pertama-tama membuat tabel dan kunci primer dalam basis data target terlebih dahulu. Kemudian gunakan tugas dengan pengaturan tugas Mode persiapan tabel target diatur ke Tidak melakukan apa pun atau Memotong untuk mengisi tabel target.

Kesalahan: Set karakter yang tidak didukung menyebabkan konversi data bidang gagal

Kesalahan berikut terjadi ketika set karakter yang tidak didukung menyebabkan konversi data bidang gagal:

```
"[SOURCE_CAPTURE ]E: Column 'column-name' uses an unsupported character set [120112]  
A field data conversion failed. (mysql_endpoint_capture.c:2154)
```

Periksa parameter database Anda yang terkait dengan koneksi. Perintah berikut dapat digunakan untuk mengatur parameter ini.

```
SHOW VARIABLES LIKE '%char%';
```

Kesalahan: Codepage 1252 ke UTF8 [120112] konversi data bidang gagal

Kesalahan berikut dapat terjadi selama migrasi jika Anda memiliki karakter non codepage-1252 di basis data MySQL sumber.

```
[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)
```

Sebagai solusi, Anda dapat menggunakan atribut koneksi tambahan CharSetMapping dengan titik akhir MySQL sumber Anda untuk menentukan pemetaan set karakter. Anda mungkin perlu memulai ulang tugas AWS DMS migrasi dari awal jika Anda menambahkan pengaturan titik akhir ini.

Misalnya, pengaturan titik akhir berikut dapat digunakan untuk titik akhir sumber MySQL di mana kumpulan karakter sumber Utf8 adalah latin1 atau 65001 adalah pengidentifikasi halaman kode UTF8.

```
CharsetMapping=utf8,65001  
CharsetMapping=latin1,65001
```

Indeks, Kunci Asing, atau Pembaruan atau Penghapusan Cascade Tidak Dimigrasi

AWS DMS tidak mendukung migrasi objek sekunder seperti indeks dan kunci asing. Untuk mereplikasi perubahan yang dibuat pada tabel turunan dari operasi pembaruan atau penghapusan kaskade, Anda harus mengaktifkan batasan kunci asing pemicu pada tabel target. Untuk mengatasi batasan ini, buat kunci asing secara manual pada tabel target. Kemudian, buat satu tugas untuk beban penuh dan CDC, atau dua tugas terpisah untuk beban penuh dan CDC, seperti yang dijelaskan berikut:

Buat satu tugas yang mendukung beban penuh dan CDC

Prosedur ini menjelaskan cara memigrasikan kunci dan indeks asing menggunakan satu tugas untuk beban penuh dan CDC.

Buat beban penuh dan tugas CDC

1. Buat tabel secara manual dengan kunci asing dan indeks pada target agar sesuai dengan tabel sumber.
2. Tambahkan ECA berikut ke titik AWS DMS akhir target:

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

3. Buat AWS DMS tugas dengan `TargetTablePrepMode` set ke `DO_NOTHING`.
4. Atur pengaturan `Stop task after full load completes` ke `StopTaskCachedChangesApplied`.
5. Mulai tugas. AWS DMS menghentikan tugas secara otomatis setelah menyelesaikan beban penuh dan menerapkan perubahan yang di-cache.
6. Hapus `SET FOREIGN_KEY_CHECKS` ECA yang Anda tambahkan sebelumnya.
7. Lanjutkan tugas. Tugas memasuki fase CDC dan menerapkan perubahan berkelanjutan dari database sumber ke target.

Buat tugas beban penuh dan CDC secara terpisah

Prosedur ini menjelaskan cara memigrasikan kunci dan indeks asing menggunakan tugas terpisah untuk beban penuh dan CDC.

Buat tugas beban penuh

1. Buat tabel secara manual dengan kunci asing dan indeks pada target agar sesuai dengan tabel sumber.
2. Tambahkan ECA berikut ke titik AWS DMS akhir target:

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

3. Buat AWS DMS tugas dengan `TargetTablePrepMode` parameter yang disetel ke `DO_NOTHING` dan `EnableValidation` atur ke `FALSE`.
4. Mulai tugas. AWS DMS menghentikan tugas secara otomatis setelah menyelesaikan beban penuh dan menerapkan perubahan yang di-cache.
5. Setelah tugas selesai, perhatikan waktu mulai tugas pemuatan penuh di UTC, atau nama dan posisi file log biner, untuk memulai tugas CDC saja. Lihat log untuk mendapatkan stempel waktu di UTC dari waktu mulai pemuatan penuh awal.

Buat tugas khusus CDC

1. Hapus `SET FOREIGN_KEY_CHECKS` ECA yang Anda atur sebelumnya.
2. Buat tugas khusus CDC dengan posisi awal yang disetel ke waktu mulai beban penuh yang dicatat pada langkah sebelumnya. Atau, Anda dapat menggunakan posisi log biner yang direkam pada langkah sebelumnya. Atur pengaturan `TargetTablePrepMode` ke `DO_NOTHING`. Aktifkan validasi data dengan `EnableValidation` menyetel pengaturan `TRUE` jika diperlukan.
3. Mulai tugas khusus CDC, dan pantau log untuk kesalahan.

Note

Solusi ini hanya berlaku untuk migrasi MySQL ke MySQL. Anda tidak dapat menggunakan metode ini dengan fitur Batch Apply, karena Batch Apply mengharuskan tabel target tidak memiliki kunci asing aktif.

Pemecahan masalah dengan PostgreSQL

Berikut ini, Anda dapat mempelajari tentang pemecahan masalah yang spesifik pada penggunaan AWS DMS dengan basis data PostgreSQL.

Topik

- [Jenis data JSON dipotong](#)
- [Kolom dari tipe data yang ditetapkan pengguna tidak dimigrasi dengan benar](#)
- [Kesalahan: Tidak ada skema yang dipilih untuk dibuat di dalamnya](#)
- [Penghapusan dan pembaruan ke tabel tidak direplikasi menggunakan CDC](#)
- [Pernyataan memotong tidak disebarakan](#)
- [Mencegah PostgreSQL dari menangkap DDL](#)
- [Memilih skema di mana objek basis data untuk menangkap DDL dibuat](#)
- [Tabel Oracle hilang setelah bermigrasi ke PostgreSQL](#)
- [ReplicationSlotDiskUsage meningkat dan restart_lsn berhenti bergerak maju selama transaksi panjang, seperti beban kerja ETL](#)
- [Tugas yang menggunakan tampilan sebagai sumber tidak memiliki baris yang disalin](#)

Jenis data JSON dipotong

AWS DMS memperlakukan jenis data JSON di PostgreSQL sebagai kolom jenis data LOB. Ini berarti bahwa batasan ukuran LOB ketika Anda menggunakan mode LOB terbatas berlaku untuk data JSON.

Sebagai contoh, misalkan mode LOB terbatas diatur ke 4.096 KB. Dalam kasus ini, data JSON yang lebih besar dari 4.096 KB dipotong pada batas 4.096 KB dan gagal dalam tes validasi di PostgreSQL.

Informasi log berikut menunjukkan JSON yang terpotong karena pengaturan mode LOB terbatas dan validasi yang gagal.

```
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)
```

Kolom dari tipe data yang ditetapkan pengguna tidak dimigrasi dengan benar

Ketika mereplikasi dari sumber PostgreSQL, AWS DMS menciptakan tabel target dengan jenis data yang sama untuk semua kolom, kecuali kolom dengan jenis data yang ditetapkan pengguna. Dalam kasus tersebut, jenis data dibuat sebagai "karakter bervariasi" dalam target.

Kesalahan: Tidak ada skema yang dipilih untuk dibuat di dalamnya

Dalam beberapa kasus, Anda mungkin melihat kesalahan "SQL_ERROR SqlState: 3F000:7 Message NativeError: ERROR: no schema has selected to create in".

Kesalahan ini dapat terjadi ketika pemetaan tabel JSON Anda berisi nilai wildcard untuk skema tetapi basis data sumber tidak mendukung nilai tersebut.

Penghapusan dan pembaruan ke tabel tidak direplikasi menggunakan CDC

Operasi penghapusan dan pembaruan selama change data capture (CDC) diabaikan jika tabel sumber tidak memiliki kunci primer. AWS DMS mendukung change data capture (CDC) untuk tabel PostgreSQL dengan kunci primer.

Jika tabel tidak memiliki kunci primer, write-ahead log (WAL) tidak mencakup citra sebelumnya dari baris basis data. Dalam kasus ini, AWS DMS tidak dapat memperbaiki tabel. Untuk operasi penghapusan yang akan direplikasi, buat kunci primer pada tabel sumber.

Pernyataan memotong tidak disebarkan

Saat menggunakan change data capture (CDC), operasi MEMOTONG tidak didukung oleh AWS DMS.

Mencegah PostgreSQL dari menangkap DDL

Anda dapat mencegah titik akhir target PostgreSQL menangkap pernyataan DDL dengan menambahkan pernyataan pengaturan Endpoint berikut.

```
"CaptureDDLs": "N"
```

Memilih skema di mana objek basis data untuk menangkap DDL dibuat

Anda dapat mengontrol skema di mana objek basis data yang terkait dengan menangkap DDL dibuat. Tambahkan pernyataan pengaturan Endpoint berikut. Parameter pengaturan Endpoint tersedia di tab titik akhir sumber.

```
"DdlArtifactsSchema": "xyzddlschema"
```

Tabel Oracle hilang setelah bermigrasi ke PostgreSQL

Dalam kasus ini, tabel dan data Anda pada umumnya masih dapat diakses.

Oracle default ke nama tabel dalam huruf besar, dan PostgreSQL default ke nama tabel dalam huruf kecil. Ketika Anda melakukan migrasi dari Oracle ke PostgreSQL, kami menyarankan bahwa Anda menyediakan aturan transformasi tertentu di bawah bagian pemetaan tabel tugas Anda. Ini adalah aturan transformasi untuk mengonversi kasus nama tabel Anda.

Jika Anda memigrasi tabel Anda tanpa menggunakan aturan transformasi untuk mengonversi kasus nama tabel Anda, lampirkan nama tabel Anda dalam tanda petik ketika mereferensikan nama tersebut.

ReplicationSlotDiskUsage meningkat dan restart_lsn berhenti bergerak maju selama transaksi panjang, seperti beban kerja ETL

Ketika replikasi logis diaktifkan, jumlah maksimum perubahan yang disimpan dalam memori per transaksi adalah 4MB. Setelah itu, perubahan ditumpahkan ke disk. Sebagai hasilnya ReplicationSlotDiskUsage meningkat, dan restart_lsn tidak berlanjut sampai transaksi

selesai/dibatalkan dan rollback selesai. Karena itu adalah transaksi yang panjang, dapat memakan waktu yang lama untuk rollback.

Jadi, hindari transaksi yang berjalan lama ketika replikasi logis diaktifkan. Sebaliknya, cobalah untuk membagi transaksi menjadi beberapa transaksi yang lebih kecil.

Tugas yang menggunakan tampilan sebagai sumber tidak memiliki baris yang disalin

Untuk memigrasi tampilan, atur `table-type` ke `all` atau `view`. Untuk informasi selengkapnya, lihat [Menentukan pemilihan tabel dan transformasi aturan dari konsol](#).

Sumber yang mendukung tampilan mencakup hal berikut.

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2 LUW
- SAP Adaptive Server Enterprise (ASE)

Pemecahan masalah Microsoft SQL Server

Berikut ini, Anda dapat mempelajari tentang pemecahan masalah yang spesifik pada penggunaan AWS DMS dengan basis data Microsoft SQL Server.

Topik

- [Kesalahan menangkap perubahan untuk basis data SQL server](#)
- [Kolom identitas hilang](#)
- [Kesalahan: SQL Server tidak mendukung publikasi](#)
- [Perubahan tidak muncul di target Anda](#)
- [Tabel yang tidak seragam dipetakan di seluruh partisi](#)

Kesalahan menangkap perubahan untuk basis data SQL server

Kesalahan selama change data capture (CDC) sering dapat menunjukkan bahwa salah satu prasyarat tidak terpenuhi. Sebagai contoh, prasyarat yang paling sering diabaikan adalah backup basis data yang lengkap. Log tugas menunjukkan kelalaian ini dengan kesalahan berikut:

```
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)
```

Tinjau prasyarat yang tercantum untuk menggunakan SQL Server sebagai sumber di [Menggunakan database Microsoft SQL Server sebagai sumber AWS DMS](#).

Kolom identitas hilang

AWS DMS tidak mendukung kolom identitas ketika Anda membuat skema target. Anda harus menambahkannya setelah beban awal selesai.

Kesalahan: SQL Server tidak mendukung publikasi

Kesalahan berikut dihasilkan ketika Anda menggunakan SQL Server Express sebagai titik akhir sumber:

```
RetCode: SQL_ERROR SqlState: HY000 NativeError: 21106  
Message: This edition of SQL Server does not support publications.
```

AWS DMS saat ini tidak mendukung SQL Server Express sebagai sumber atau target.

Perubahan tidak muncul di target Anda

AWS DMS mensyaratkan bahwa basis data SQL Server sumber berada dalam model pemulihan data 'FULL' atau 'BULK LOGGED' agar bisa secara konsisten menangkap perubahan. Model 'SIMPLE' tidak didukung.

Model pemulihan SIMPLE mencatat informasi minimal yang diperlukan untuk mengizinkan pengguna memulihkan basis data mereka. Semua entri log yang tidak aktif secara otomatis terpotong ketika titik pemeriksaan terjadi.

Semua operasi masih dicatat. Namun, segera setelah titik pemeriksaan terjadi log secara otomatis terpotong. Pemotongan ini berarti bahwa log menjadi tersedia untuk digunakan kembali dan entri log yang lebih tua dapat ditimpa. Ketika entri log ditimpa, perubahan tidak dapat ditangkap. Masalah ini adalah alasan mengapa AWS DMS tidak mendukung model pemulihan data SIMPLE. Untuk informasi tentang prasyarat lain yang diperlukan untuk menggunakan SQL Server sebagai sumber, lihat [Menggunakan database Microsoft SQL Server sebagai sumber AWS DMS](#).

Tabel yang tidak seragam dipetakan di seluruh partisi

Selama change data capture (CDC), migrasi tabel dengan struktur khusus ditangguhkan ketika AWS DMS tidak dapat melakukan CDC di tabel dengan benar. Pesan seperti ini dikeluarkan:

```
[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)
```

Saat menjalankan CDC pada tabel SQL Server, AWS DMS mengurai tlog SQL Server. Pada setiap catatan tlog, AWS DMS mengurai nilai heksadesimal yang berisi data untuk kolom yang dimasukkan, diperbarui, atau dihapus selama perubahan.

Untuk mengurai catatan heksadesimal, AWS DMS membaca metadata tabel dari tabel sistem SQL Server. Tabel sistem tersebut mengidentifikasi kolom tabel terstruktur khusus tersebut dan mengungkapkan beberapa properti internal mereka, seperti "xoffset" dan "null bit position".

AWS DMS mengharapkan bahwa metadata harus sama untuk semua partisi mentah tabel. Namun dalam beberapa kasus, tabel terstruktur khusus tidak memiliki metadata yang sama pada semua partisi mereka. Dalam kasus ini, AWS DMS dapat menangguhkan CDC pada tabel tersebut untuk menghindari menguraikan perubahan dengan tidak benar dan memberikan data yang salah kepada target. Solusi mencakup hal berikut:

- Jika tabel memiliki indeks berklaster, lakukan pembuatan ulang indeks.
- Jika tabel tidak memiliki indeks berklaster, tambahkan indeks berklaster ke tabel (Anda dapat menghapusnya nanti jika Anda ingin).

Pemecahan masalah dengan Amazon Redshift

Berikut ini, Anda dapat mempelajari tentang pemecahan masalah yang spesifik pada penggunaan AWS DMS dengan basis data Amazon Redshift.

Topik

- [Memuat ke sebuah klaster Amazon Redshift di Wilayah AWS yang berbeda](#)
- [Kesalahan: Relasi "awsdms_apply_exceptions" sudah ada](#)
- [Kesalahan dengan tabel yang namanya dimulai dengan "awsdms_changes"](#)
- [Melihat tabel dalam klaster dengan nama seperti dms.awsdms_changes000000000XXXX](#)
- [Izin yang diperlukan untuk bekerja dengan Amazon Redshift](#)

Memuat ke sebuah klaster Amazon Redshift di Wilayah AWS yang berbeda

Anda tidak dapat memuat ke sebuah klaster Amazon Redshift di Wilayah AWS yang berbeda dari instans replikasi AWS DMS Anda. DMS mengharuskan instans replikasi dan klaster Amazon Redshift berada di wilayah yang sama.

Kesalahan: Relasi "awsdms_apply_exceptions" sudah ada

Kesalahan "Relasi 'awsdms_apply_exceptions' sudah ada" sering terjadi ketika titik akhir Redshift ditetapkan sebagai titik akhir PostgreSQL. Untuk mengatasi masalah ini, modifikasi titik akhir dan ubah Mesin target ke "redshift."

Kesalahan dengan tabel yang namanya dimulai dengan "awsdms_changes"

Pesan kesalahan tabel dengan nama yang dimulai dengan "awsdms_changes" dapat terjadi ketika dua tugas mencoba untuk memuat data ke dalam klaster Amazon Redshift yang sama yang berjalan secara bersamaan. Karena cara tabel sementara diberi nama, tugas yang dilakukan bersamaan dapat bertentangan ketika memperbarui tabel yang sama.

Melihat tabel dalam klaster dengan nama seperti dms.awsdms_changes000000000XXXX

AWS DMS membuat tabel sementara ketika data sedang dimuat dari file yang disimpan di Amazon S3. Nama tabel sementara ini masing-masing memiliki prefiks dms . awsdms_changes. Tabel ini

diperlukan sehingga AWS DMS dapat menyimpan data ketika pertama kali dimuat dan sebelum ditempatkan dalam tabel target akhir.

Izin yang diperlukan untuk bekerja dengan Amazon Redshift

Untuk menggunakan AWS DMS dengan Amazon Redshift, akun pengguna yang Anda gunakan untuk mengakses Amazon Redshift harus memiliki izin berikut:

- CRUD (Pilih, Sisipkan, Perbarui, Hapus)
- Beban massal
- Membuat, mengubah, membatalkan (jika diperlukan oleh definisi tugas)

Untuk melihat prasyarat yang diperlukan untuk menggunakan Amazon Redshift sebagai target, lihat [Menggunakan basis data Amazon Redshift sebagai target untuk AWS Database Migration Service](#).

Pemecahan masalah dengan Amazon Aurora MySQL

Berikut ini, Anda dapat mempelajari tentang pemecahan masalah yang spesifik pada penggunaan AWS DMS dengan basis data Amazon Aurora MySQL.

Topik

- [Kesalahan: SET KARAKTER bidang UTF8 diakhiri oleh ',' diapit oleh "" baris diakhiri oleh '\n'](#)

Kesalahan: SET KARAKTER bidang UTF8 diakhiri oleh ',' diapit oleh "" baris diakhiri oleh '\n'

Jika Anda menggunakan Amazon Aurora MySQL sebagai target, Anda mungkin melihat kesalahan seperti berikut ini di log. Jenis kesalahan ini biasanya menunjukkan bahwa Anda memiliki ANSI_QUOTES sebagai bagian dari parameter SQL_MODE. Memiliki ANSI_QUOTES sebagai bagian dari parameter SQL_MODE menyebabkan tanda kutip ganda untuk ditangani seperti tanda kutip dan dapat membuat masalah ketika Anda menjalankan tugas.

Untuk memperbaiki kesalahan ini, hapus ANSI_QUOTES dari parameter SQL_MODE.

```
2016-11-02T14:23:48 [TARGET_LOAD ]E: Load data sql statement. load data local infile
```

```
"/rdsdbdata/data/tasks/7X04FJHCV0N7TYTLQ6RX3CQH DU/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)
```

Memecahkan masalah dengan SAP ASE

Berikut ini, Anda dapat mempelajari tentang pemecahan masalah khusus untuk digunakan AWS DMS dengan database SAP ASE.

Kesalahan: Kolom LOB memiliki nilai NULL ketika sumber memiliki indeks unik komposit dengan nilai NULL

Saat menggunakan SAP ASE sebagai sumber dengan tabel yang dikonfigurasi dengan indeks unik komposit yang memungkinkan nilai NULL, nilai LOB mungkin tidak bermigrasi selama replikasi yang sedang berlangsung. Perilaku ini biasanya merupakan hasil dari ANSI_NULL disetel ke 1 secara default pada klien instance replikasi DMS.

Untuk memastikan bahwa bidang LOB bermigrasi dengan benar, sertakan pengaturan Endpoint 'AnsiNull=0' ke titik akhir AWS DMS sumber untuk tugas tersebut.

Memecahkan masalah dengan IBM Db2

Berikut ini, Anda dapat mempelajari tentang pemecahan masalah khusus untuk menggunakan AWS DMS dengan database IBM Db2.

Kesalahan: Lanjutkan dari stempel waktu tidak didukung Tugas

Untuk replikasi berkelanjutan (CDC), jika Anda berencana untuk memulai replikasi dari stempel waktu tertentu, setel atribut `StartFromContext` koneksi ke stempel waktu yang diperlukan. Untuk informasi selengkapnya, lihat [Pengaturan titik akhir saat menggunakan Db2 LUW](#). Pengaturan `StartFromContext` ke stempel waktu yang diperlukan mencegah masalah berikut:

```
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.
```

Memecahkan masalah latensi di AWS Database Migration Service

Bagian ini memberikan gambaran umum tentang penyebab umum latensi AWS DMS tugas selama fase replikasi yang sedang berlangsung (CDC). AWS DMS mereplikasi data secara asinkron. Latensi adalah penundaan antara saat perubahan dilakukan pada sumber dan ketika perubahan direplikasi ke target. Latensi dapat disebabkan karena kesalahan konfigurasi komponen replikasi, seperti berikut ini:

- Titik akhir sumber atau sumber data
- Titik akhir target atau sumber data
- Instans replikasi
- Jaringan antara komponen-komponen ini

Kami menyarankan Anda menggunakan migrasi uji sebagai bukti konsep untuk mengumpulkan informasi tentang replikasi Anda. Anda kemudian dapat menggunakan informasi ini untuk menyetel konfigurasi replikasi Anda untuk meminimalkan latensi. Untuk informasi tentang menjalankan bukti migrasi konsep, lihat [Menjalankan bukti konsep](#).

Topik

- [Jenis latensi CDC](#)
- [Penyebab umum latensi CDC](#)
- [Memecahkan masalah latensi](#)

Jenis latensi CDC

Bagian ini berisi jenis latensi replikasi yang mungkin terjadi selama CDC.

Latensi sumber

Penundaan, dalam hitungan detik, antara waktu komit dari peristiwa terakhir yang diambil dari titik akhir sumber, dan stempel waktu sistem saat ini dari instance replikasi. Anda dapat memantau

latensi antara sumber data dan instance replikasi Anda menggunakan metrik. `CDCLatencySource` CloudWatch `CDCLatencySource` metrik tinggi menunjukkan bahwa proses menangkap perubahan dari sumber tertunda. Misalnya, jika aplikasi Anda melakukan insert ke sumber pada 10:00, dan AWS DMS mengkonsumsi perubahan pada 10:02, metriknya adalah 120 detik. `CDCLatencySource`

Untuk informasi tentang CloudWatch metrik AWS DMS, lihat [Metrik tugas replikasi](#).

Target latensi

Penundaan, dalam hitungan detik, antara waktu komit pada sumber peristiwa pertama yang menunggu untuk berkomitmen ke target, dan stempel waktu saat ini dari instance replikasi DMS. Anda dapat memantau latensi antara komit pada sumber data dan target data Anda menggunakan metrik. `CDCLatencyTarget` CloudWatch Ini berarti itu `CDCLatencyTarget` termasuk keterlambatan membaca dari sumbernya. Akibatnya, selalu `CDCLatencyTarget` lebih besar dari atau sama dengan `CDCLatencySource`.

Misalnya, jika aplikasi Anda melakukan penyisipan ke sumber pada pukul 10:00, dan menggunakannya pada pukul 10:02 dan AWS DMS menuliskannya ke target pada pukul 10:05, metriknya adalah 300 detik. `CDCLatencyTarget`

Penyebab umum latensi CDC

Bagian ini berisi penyebab latensi yang mungkin dialami replikasi Anda selama CDC.

Topik

- [Sumber daya titik akhir](#)
- [Sumber daya contoh replikasi](#)
- [Kecepatan jaringan dan bandwidth](#)
- [Konfigurasi DMS](#)
- [Skenario replikasi](#)

Sumber daya titik akhir

Faktor-faktor berikut secara signifikan mempengaruhi kinerja replikasi dan latensi:

- Konfigurasi basis data sumber dan target
- Ukuran instans

- Penyimpanan data sumber atau target yang kurang disediakan atau salah konfigurasi

Untuk mengidentifikasi penyebab latensi yang disebabkan oleh masalah titik akhir untuk sumber dan target yang AWS di-host, pantau metrik berikut: CloudWatch

- FreeMemory
- CPUUtilization
- Metrik throughput dan I/O, seperti WriteIOPS, atau WriteThroughput ReadLatency
- Metrik volume transaksi seperti CDCIncomingChanges.

Untuk informasi tentang CloudWatch metrik pemantauan, lihat [Metrik AWS Database Migration Service](#).

Sumber daya contoh replikasi

Sumber daya instans replikasi sangat penting untuk replikasi, dan Anda harus memastikan bahwa tidak ada hambatan sumber daya, karena dapat menyebabkan latensi sumber dan target.

Untuk mengidentifikasi kemacetan sumber daya untuk instance replikasi Anda, verifikasi hal berikut:

- CloudWatch Metrik penting seperti CPU, Memori, I/O per detik, dan penyimpanan tidak mengalami lonjakan atau nilai tinggi yang konsisten.
- Instance replikasi Anda berukuran tepat untuk beban kerja Anda. Untuk informasi tentang menentukan ukuran yang benar dari instance replikasi, lihat [Memilih ukuran terbaik untuk contoh replikasi](#).

Kecepatan jaringan dan bandwidth

Bandwith jaringan adalah faktor yang mempengaruhi transmisi data. Untuk menganalisis kinerja jaringan replikasi Anda, lakukan salah satu hal berikut:

- Periksa ReadThroughput dan WriteThroughput metrik di tingkat instans. Untuk informasi tentang CloudWatch metrik pemantauan, lihat [Metrik AWS Database Migration Service](#).
- Gunakan Dukungan AWS DMS Diagnostik AMI. Jika Dukungan Diagnostik AMI tidak tersedia di wilayah Anda, Anda dapat mengunduhnya dari wilayah mana pun yang didukung dan menyalinnya ke wilayah Anda untuk melakukan analisis jaringan Anda. Untuk informasi tentang Dukungan Diagnostik AMI, lihat [Bekerja dengan dukungan AWS DMS diagnostik AMI](#).

CDC in AWS DMS adalah single-threaded untuk memastikan konsistensi data. Akibatnya, Anda dapat menentukan volume data yang dapat didukung jaringan Anda dengan menghitung kecepatan transfer data single-threaded Anda. Misalnya, jika tugas Anda terhubung ke sumbernya menggunakan jaringan 100 Mbps (megabit per detik), replikasi Anda memiliki alokasi bandwidth maksimum teoritis 12,5 MBps (megabyte per detik). Ini sama dengan 45 gigabit per jam. Jika tingkat pembuatan log transaksi pada sumber lebih besar dari 45 gigabit per jam, ini berarti bahwa tugas tersebut memiliki latensi CDC. Untuk jaringan 100 MBps, tingkat ini adalah maksimum teoritis; faktor lain seperti lalu lintas jaringan dan overhead sumber daya pada sumber dan target mengurangi bandwidth yang tersedia aktual.

Konfigurasi DMS

Bagian ini berisi konfigurasi replikasi yang direkomendasikan yang dapat membantu mengurangi latensi.

- Pengaturan titik akhir: Pengaturan titik akhir sumber dan target Anda dapat menyebabkan instance replikasi Anda mengalami kinerja yang buruk. Pengaturan titik akhir yang mengaktifkan fitur intensif sumber daya akan memengaruhi kinerja. Misalnya, untuk titik akhir Oracle, menonaktifkan LogMiner dan menggunakan Binary Reader meningkatkan kinerja, karena intensif sumber daya. LogMiner Pengaturan endpoing berikut meningkatkan kinerja untuk endpoint Oracle:

```
useLogminerReader=N;useBfile=Y
```

Untuk informasi selengkapnya tentang pengaturan titik akhir, lihat dokumentasi untuk mesin sumber dan titik akhir target Anda dalam topik [Bekerja dengan Titik akhir DMS AWS](#)

- Pengaturan tugas: Beberapa pengaturan tugas untuk skenario replikasi khusus Anda dapat menyebabkan instance replikasi Anda mengalami kinerja yang buruk. Misalnya, AWS DMS menggunakan mode penerapan transaksional secara default (`BatchApplyEnabled=false`) untuk CDC untuk semua titik akhir kecuali Amazon Redshift. Namun, untuk sumber dengan sejumlah besar perubahan, pengaturan `BatchApplyEnabled` untuk `true` dapat meningkatkan kinerja.

Untuk informasi selengkapnya tentang pengaturan tugas, lihat [Menentukan setelan tugas untuk tugas AWS Database Migration Service](#).

- Mulai Posisi tugas hanya CDC: Memulai tugas khusus CDC dari posisi atau stempel waktu di masa lalu akan memulai tugas dengan peningkatan latensi sumber CDC. Bergantung pada volume perubahan pada sumber, latensi tugas akan membutuhkan waktu untuk mereda.

- Pengaturan LOB: Tipe data Objek Besar dapat menghambat kinerja replikasi karena cara AWS DMS mereplikasi data biner besar. Untuk informasi selengkapnya, lihat topik berikut:
 - [Mengatur dukungan LOB untuk database sumber dalam suatu tugas AWS DMS](#)
 - [Melakukan migrasi objek biner large \(LOB\)](#).

Skenario replikasi

Bagian ini menjelaskan skenario replikasi spesifik dan bagaimana mereka dapat mempengaruhi latensi.

Topik

- [Menghentikan tugas untuk jangka waktu yang lama](#)
- [Perubahan cache](#)
- [Replikasi lintas wilayah](#)

Menghentikan tugas untuk jangka waktu yang lama

Ketika Anda menghentikan tugas, AWS DMS menyimpan posisi log transaksi terakhir yang dibaca dari sumbernya. Ketika Anda melanjutkan tugas, DMS mencoba untuk melanjutkan membaca dari posisi log transaksi yang sama. Melanjutkan tugas setelah beberapa jam atau hari menyebabkan latensi sumber CDC meningkat hingga DMS selesai menghabiskan backlog transaksi.

Perubahan cache

Perubahan cache adalah perubahan yang ditulis aplikasi Anda ke sumber data saat AWS DMS menjalankan fase replikasi beban penuh. DMS tidak menerapkan perubahan ini sampai fase beban penuh selesai dan fase CDC dimulai. Untuk sumber dengan sejumlah besar transaksi, perubahan cache membutuhkan waktu lebih lama untuk diterapkan, sehingga latensi sumber meningkat ketika fase CDC dimulai. Kami menyarankan Anda menjalankan fase beban penuh ketika volume transaksi rendah untuk meminimalkan jumlah perubahan yang di-cache.

Replikasi lintas wilayah

Menemukan titik akhir DMS atau instans replikasi Anda di berbagai AWS wilayah meningkatkan latensi jaringan. Ini meningkatkan latensi replikasi. Untuk performa terbaik, cari titik akhir sumber, titik akhir target, dan instance replikasi Anda di wilayah yang sama. AWS

Memecahkan masalah latensi

Bagian ini berisi langkah-langkah pemecahan masalah untuk latensi replikasi.

Untuk memecahkan masalah latensi, lakukan hal berikut:

- Pertama, tentukan jenis dan jumlah latensi untuk tugas tersebut. Periksa bagian Statistik Tabel tugas dari konsol DMS atau CLI. Jika penghitung berubah, maka transmisi data sedang berlangsung. Periksa `CDCLatencySource` dan `CDCLatencyTarget` metrik bersama-sama untuk menentukan apakah ada hambatan selama CDC.
- Jika tinggi `CDCLatencySource` atau `CDCLatencyTarget` metrik menunjukkan hambatan dalam replikasi Anda, periksa hal berikut:
 - Jika `CDCLatencySource` tinggi dan `CDCLatencyTarget` sama dengan `CDCLatencySource`, ini menunjukkan bahwa ada hambatan di titik akhir sumber Anda, dan menulis data ke AWS DMS target dengan lancar. Lihat [Memecahkan masalah latensi sumber](#) berikut.
 - Jika `CDCLatencySource` rendah dan `CDCLatencyTarget` tinggi, ini menunjukkan bahwa ada hambatan di titik akhir target Anda, dan AWS DMS membaca data dari sumber dengan lancar. Lihat [Memecahkan masalah latensi target](#) berikut.
 - Jika `CDCLatencySource` tinggi dan secara signifikan `CDCLatencyTarget` lebih tinggi dari `CDCLatencySource`, ini menunjukkan kemacetan pada pembacaan sumber dan penulisan target. Selidiki latensi sumber terlebih dahulu, lalu selidiki latensi target.

Untuk informasi tentang memantau metrik tugas DMS, lihat. [Pemantauan Tugas AWS DMS](#)

Memecahkan masalah latensi sumber

Topik berikut menjelaskan skenario replikasi khusus untuk jenis titik akhir sumber.

Topik

- [Pemecahan Masalah Oracle Endpoint](#)
- [Pemecahan Masalah Titik Akhir MySQL](#)
- [Pemecahan Masalah PostgreSQL Endpoint](#)
- [Pemecahan Masalah Titik Akhir SQL Server](#)

Pemecahan Masalah Oracle Endpoint

Bagian ini berisi skenario replikasi khusus untuk Oracle.

Pembacaan sumber dijeda

AWS DMS berhenti membaca dari sumber Oracle dalam skenario berikut. Perilaku ini adalah dengan desain. Anda dapat menyelidiki penyebabnya menggunakan log tugas. Cari pesan yang mirip dengan yang berikut ini di log tugas. Untuk informasi tentang bekerja dengan log tugas, lihat [Melihat dan mengelola AWS Mencatat tugas DMS](#).

- Pesan SORTER: Ini menunjukkan bahwa DMS melakukan caching transaksi pada instance replikasi. Untuk informasi lebih lanjut, lihat [Pesan SORTER di log tugas](#) berikut ini.
- Log tugas debug: Jika DMS mengganggu proses baca, tugas Anda berulang kali menulis pesan berikut ke log tugas debug, tanpa mengubah bidang konteks atau stempel waktu:
 - Pembaca biner:

```
[SOURCE_CAPTURE ]T: Produce CTI event:
context '00000020.f23ec6e5.00000002.000a.00.0000:190805.3477731.16'
xid [0000000001e0018] 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
(oradcdc_reader.c:2269)
```

- AWS DMS mencatat pesan berikut untuk setiap pengulangan baru atau operasi log yang diarsipkan.

```
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)
```

Jika sumber memiliki pengulangan baru atau operasi log yang diarsipkan, dan AWS DMS tidak menulis pesan ini ke log, ini berarti bahwa tugas tersebut tidak memproses peristiwa.

Generasi redo tinggi

Jika tugas Anda memproses log redo atau diarsipkan, tetapi latensi sumber tetap tinggi, coba identifikasi tingkat pembuatan log ulang dan pola pembuatan. Jika Anda memiliki tingkat pembuatan

redo log yang tinggi, ini meningkatkan latensi sumber, karena tugas Anda membaca semua log redo dan arsip untuk mengambil perubahan yang terkait dengan tabel yang direplikasi.

Untuk menentukan tingkat pembuatan ulang, gunakan kueri berikut.

- Tingkat pembuatan ulang per hari:

```
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;
```

- Tingkat pembuatan ulang per jam:

```
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 ;
```

Untuk memecahkan masalah latensi dalam skenario ini, periksa hal berikut:

- Periksa bandwidth jaringan dan kinerja single-thread replikasi Anda untuk memastikan bahwa jaringan yang mendasari Anda dapat mendukung tingkat pembuatan ulang sumber. Untuk informasi tentang bagaimana bandwidth jaringan dapat mempengaruhi kinerja replikasi, lihat [Kecepatan jaringan dan bandwidth](#) sebelumnya.
- Periksa apakah Anda mengatur pencatatan tambahan dengan benar. Hindari pencatatan tambahan pada sumber, seperti mengaktifkan logging pada semua kolom tabel. Untuk informasi tentang menyiapkan pencatatan tambahan, lihat. [Mengatur supplemental logging](#)
- Verifikasi bahwa Anda menggunakan API yang benar untuk membaca log redo atau archived. Anda dapat menggunakan Oracle LogMiner atau AWS DMS Binary Reader. Saat LogMiner membaca log redo online dan file log redo yang diarsipkan, Binary Reader membaca dan mem-parsing file log redo mentah secara langsung. Akibatnya, Binary Reader lebih berkinerja. Kami menyarankan Anda menggunakan Binary Reader jika pembuatan redo log Anda lebih dari 10 GB/jam. Untuk informasi selengkapnya, lihat [Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC](#).

- Periksa apakah Anda mengatur `ArchivedLogsOnly` ke `Y`. Jika pengaturan titik akhir ini disetel, AWS DMS membaca dari log pengulangan yang diarsipkan. Ini meningkatkan latensi sumber, karena AWS DMS menunggu log pengulangan online diarsipkan sebelum membaca. Untuk informasi lebih lanjut, lihat [ArchivedLogsOnly](#).
- Jika sumber Oracle Anda menggunakan Automatic Storage Management (ASM), lihat [Menyimpan REDO di Oracle ASM saat menggunakan Oracle sebagai sumber AWS DMS](#) informasi tentang cara mengonfigurasi penyimpanan data dengan benar. Anda mungkin juga dapat mengoptimalkan kinerja membaca lebih lanjut dengan menggunakan attribute koneksi `asmUsePLSQLArray` tambahan (ECA). Untuk informasi tentang penggunaan `asmUsePLSQLArray`, lihat [Pengaturan titik akhir saat menggunakan Oracle sebagai sumber AWS DMS](#).

Pemecahan Masalah Titik Akhir MySQL

Bagian ini berisi skenario replikasi khusus untuk MySQL. AWS DMS memindai log biner MySQL secara berkala untuk mereplikasi perubahan. Proses ini dapat meningkatkan latensi dalam skenario berikut:

Topik

- [Transaksi jangka panjang pada sumber](#)
- [Beban kerja yang tinggi pada sumbernya](#)
- [Perdebatan log biner](#)

Transaksi jangka panjang pada sumber

Karena MySQL hanya menulis transaksi yang berkomitmen ke log biner, transaksi yang berjalan lama menyebabkan lonjakan latensi sebanding dengan waktu proses kueri.

Untuk mengidentifikasi transaksi yang berjalan lama, gunakan kueri berikut, atau gunakan log kueri lambat:

```
SHOW FULL PROCESSLIST;
```

Untuk informasi tentang menggunakan log kueri lambat, lihat [Log Kueri Lambat](#) dalam dokumentasi [MySQL](#).

Untuk menghindari lonjakan latensi dari transaksi yang berjalan lama, restrukturisasi transaksi sumber Anda untuk mengurangi waktu proses kueri atau meningkatkan frekuensi komit Anda.

Beban kerja yang tinggi pada sumbernya

Karena DMS CDC adalah single-threaded, sejumlah besar transaksi dapat meningkatkan latensi sumber. Untuk mengidentifikasi apakah latensi sumber disebabkan oleh beban kerja yang berat, bandingkan jumlah dan ukuran log biner yang dihasilkan selama periode latensi dengan log yang dihasilkan sebelum latensi. Untuk memeriksa log biner, dan status utas DMS CDC, gunakan kueri berikut:

```
SHOW BINARY LOGS;  
SHOW PROCESSLIST;
```

Untuk informasi selengkapnya tentang status utas dump log biner CDC, lihat Status Thread [Sumber Replikasi](#).

Anda dapat menentukan latensi dengan membandingkan posisi log biner terbaru yang dihasilkan pada sumber dengan peristiwa yang sedang diproses DMS. Untuk mengidentifikasi log biner terbaru pada sumbernya, lakukan hal berikut:

- Aktifkan log debug pada komponen SOURCE_CAPTURE.
- Ambil log biner pemrosesan DMS dan detail posisi dari log debug tugas.
- Gunakan kueri berikut untuk mengidentifikasi log biner terbaru pada sumbernya:

```
SHOW MASTER STATUS;
```

Untuk lebih mengoptimalkan kinerja, sesuaikan `EventsPollInterval`. Secara default, DMS melakukan polling log biner setiap 5 detik, tetapi Anda dapat meningkatkan kinerja dengan mengurangi nilai ini. Untuk informasi selengkapnya tentang pengaturan `EventsPollInterval`, lihat [Pengaturan titik akhir saat menggunakan MySQL sebagai sumber AWS DMS](#).

Perdebatan log biner

Saat memigrasikan beberapa tabel dengan sejumlah besar data, sebaiknya pisahkan tabel menjadi tugas terpisah untuk MySQL 5.7.2 atau yang lebih baru. Di MySQL versi 5.7.2 dan yang lebih baru, thread dump master menciptakan lebih sedikit kontensi kunci dan meningkatkan throughput. Akibatnya, utas dump tidak lagi mengunci log biner setiap kali membaca suatu peristiwa. Ini berarti bahwa beberapa utas dump dapat membaca file log biner secara bersamaan. Ini juga berarti bahwa dump thread dapat membaca log biner sementara klien menulis ke sana. [Untuk informasi selengkapnya tentang dump thread, lihat Replication Threads dan catatan rilis MySQL 5.7.2.](#)

Untuk meningkatkan kinerja replikasi untuk versi sumber MySQL sebelum 5.7.2, coba konsolidasikan tugas dengan komponen CDC.

Pemecahan Masalah PostgreSQL Endpoint

Bagian ini berisi skenario replikasi khusus untuk PostgreSQL.

Topik

- [Transaksi jangka panjang pada sumber](#)
- [Beban kerja yang tinggi pada sumbernya](#)
- [Throughput jaringan tinggi](#)
- [Menumpahkan file di Aurora PostgreSQL](#)

Transaksi jangka panjang pada sumber

Ketika ada transaksi jangka panjang dalam database sumber, seperti beberapa ribu sisipan dalam satu transaksi, peristiwa DMS CDC dan penghitung transaksi tidak meningkat sampai transaksi selesai. Penundaan ini dapat menyebabkan masalah latensi yang dapat Anda ukur menggunakan `CDCLatencyTarget` metrik.

Untuk meninjau transaksi jangka panjang, lakukan salah satu hal berikut:

- Gunakan `pg_replication_slots` tampilan. Jika `restart_lsn` nilainya tidak diperbarui, kemungkinan PostgreSQL tidak dapat menulis Write Ahead Logs (WALS) karena transaksi aktif yang berjalan lama. [Untuk informasi tentang `pg_replication_slots` tampilan, lihat `pg_replication_slots` di Dokumentasi PostgreSQL 15.4.](#)
- Gunakan kueri berikut untuk mengembalikan daftar semua kueri aktif dalam database, bersama dengan informasi terkait:

```
SELECT pid, age(clock_timestamp(), query_start), username, query
FROM pg_stat_activity WHERE query != '<IDLE>'
AND query NOT ILIKE '%pg_stat_activity%'
ORDER BY query_start desc;
```

Dalam hasil kueri, `age` bidang menunjukkan durasi aktif setiap kueri, yang dapat Anda gunakan untuk mengidentifikasi kueri yang berjalan lama.

Beban kerja yang tinggi pada sumbernya

Jika PostgreSQL sumber Anda memiliki beban kerja yang tinggi, periksa hal berikut untuk mengurangi latensi:

- Anda mungkin mengalami latensi tinggi saat menggunakan `test_decoding` plugin saat memigrasikan subset tabel dari database sumber dengan nilai transaksi per detik (TPS) tinggi. Ini karena `test_decoding` plugin mengirimkan semua perubahan database ke contoh replikasi yang kemudian difilter DMS, berdasarkan pemetaan tabel tugas. Peristiwa untuk tabel yang bukan bagian dari pemetaan tabel tugas dapat meningkatkan latensi sumber.
- Periksa throughput TPS menggunakan salah satu metode berikut.
 - Untuk sumber Aurora PostgreSQL, gunakan metrik `CommitThroughput` CloudWatch
 - Untuk PostgreSQL yang berjalan di Amazon RDS atau lokal, gunakan kueri berikut menggunakan klien PSQL versi 11 atau yang lebih tinggi **enter** (Tekan selama kueri untuk memajukan hasil):

```
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";
```

- Untuk mengurangi latensi saat menggunakan `test_decoding` plugin, pertimbangkan untuk menggunakan `pglogical` plugin sebagai gantinya. Berbeda dengan `test_decoding` plugin, filter `pglogical` plugin write ahead log (WAL) berubah di sumbernya, dan hanya mengirimkan perubahan yang relevan ke instance replikasi. Untuk informasi tentang menggunakan `pglogical` plugin dengan AWS DMS, lihat [Mengonfigurasi plugin pglogical](#).

Throughput jaringan tinggi

Replikasi Anda mungkin memiliki penggunaan bandwidth jaringan yang tinggi saat menggunakan `test_decoding` plugin, terutama selama transaksi volume tinggi. Ini karena `test_decoding` plugin memproses perubahan, dan mengubahnya menjadi format yang dapat dibaca manusia yang lebih besar dari format biner asli.

Untuk meningkatkan kinerja, pertimbangkan untuk menggunakan `pglogical` plugin sebagai gantinya, yang merupakan plugin biner. Berbeda dengan `test_decoding` plugin, `pglogical` plugin menghasilkan output format biner, menghasilkan perubahan aliran write ahead log (WAL) terkompresi.

Menumpahkan file di Aurora PostgreSQL

Dalam PostgreSQL versi 13 dan lebih tinggi, parameter menentukan alokasi memori `logical_decoding_work_mem` untuk decoding dan streaming. [Untuk informasi selengkapnya tentang `logical_decoding_work_mem` parameter, lihat Konsumsi Sumber Daya di PostgreSQL di Dokumentasi PostgreSQL 13.13.](#)

Replikasi logis mengakumulasi perubahan untuk semua transaksi dalam memori sampai transaksi tersebut dilakukan. Jika jumlah data yang disimpan di semua transaksi melebihi jumlah yang ditentukan oleh parameter database `logical_decoding_work_mem`, maka DMS menumpahkan data transaksi ke disk untuk melepaskan memori untuk data decoding baru.

Transaksi yang berjalan lama, atau banyak subtransaksi, dapat mengakibatkan DMS mengkonsumsi memori decoding logis yang meningkat. Peningkatan penggunaan memori ini menghasilkan DMS membuat file tumpahan pada disk, yang menyebabkan latensi sumber tinggi selama replikasi.

Untuk mengurangi dampak peningkatan beban kerja sumber, lakukan hal berikut:

- Kurangi transaksi yang berjalan lama.
- Mengurangi jumlah sub-transaksi.
- Hindari melakukan operasi yang menghasilkan ledakan besar catatan log, seperti menghapus atau memperbarui seluruh tabel dalam satu transaksi. Lakukan operasi dalam batch yang lebih kecil sebagai gantinya.

Anda dapat menggunakan CloudWatch metrik berikut untuk memantau beban kerja pada sumbernya:

- `TransactionLogsDiskUsage`: Jumlah byte yang saat ini ditempati oleh WAL logis. Nilai ini meningkat secara monoton jika slot replikasi logis tidak dapat mengikuti laju penulisan baru, atau jika ada transaksi yang berjalan lama mencegah pengumpulan sampah file lama.
- `ReplicationSlotDiskUsage`: Jumlah ruang disk yang digunakan slot replikasi logis saat ini.

Anda dapat mengurangi latensi sumber dengan menyetel parameter.

`logical_decoding_work_mem` Nilai default untuk parameter ini adalah 64 MB. Parameter ini membatasi jumlah memori yang digunakan oleh setiap koneksi replikasi streaming logis. Kami merekomendasikan pengaturan `logical_decoding_work_mem` nilai secara signifikan lebih tinggi daripada `work_mem` nilai untuk mengurangi jumlah perubahan yang diterjemahkan yang ditulis DMS ke disk.

Kami menyarankan Anda memeriksa file tumpahan secara berkala, terutama selama periode aktivitas migrasi berat atau latensi. Jika DMS membuat sejumlah besar file tumpahan, ini berarti bahwa decoding logis tidak beroperasi secara efisien, yang dapat meningkatkan latensi. Untuk mengurangi ini, tingkatkan nilai `logical_decoding_work_mem` parameter.

Anda dapat memeriksa overflow transaksi saat ini dengan `aurora_stat_file` fungsi tersebut. Untuk informasi selengkapnya, lihat [Menyesuaikan memori kerja untuk decoding logis](#) di Panduan Pengembang Layanan Amazon Relational Database Service.

Pemecahan Masalah Titik Akhir SQL Server

Bagian ini berisi skenario replikasi khusus untuk SQL Server. Untuk menentukan perubahan apa yang akan direplikasi dari SQL server AWS DMS membaca log transaksi, dan menjalankan pemindaian berkala pada database sumber. Latensi replikasi biasanya dihasilkan dari SQL Server yang membatasi pemindaian ini karena kendala sumber daya. Ini juga dapat dihasilkan dari peningkatan yang signifikan dalam jumlah peristiwa yang ditulis ke log transaksi dalam waktu singkat.

Topik

- [Indeks membangun kembali](#)
- [Transaksi besar](#)
- [Interval polling MS-CDC yang salah dikonfigurasi untuk Amazon RDS SQL Server](#)
- [Beberapa tugas CDC mereplikasi dari database sumber yang sama](#)

Indeks membangun kembali

Ketika SQL Server membangun kembali indeks besar, ia menggunakan satu transaksi. Ini menghasilkan banyak peristiwa, dan dapat menggunakan sejumlah besar ruang log jika SQL Server membangun kembali beberapa indeks sekaligus. Ketika ini terjadi, Anda dapat mengharapkan lonjakan replikasi singkat. Jika sumber SQL Server Anda memiliki lonjakan log yang berkelanjutan, periksa hal berikut:

- Pertama, periksa periode waktu lonjakan latensi menggunakan `CDCLatencySource` CloudWatch metrik `CDCLatencySource` dan, atau dengan memeriksa pesan Pemantauan Throughput di log tugas. Untuk informasi tentang CloudWatch metrik AWS DMS, lihat [Metrik tugas replikasi](#).
- Periksa apakah ukuran log transaksi aktif atau cadangan log meningkat selama lonjakan latensi. Periksa juga apakah pekerjaan pemeliharaan atau pembangunan kembali berjalan selama waktu

itu. Untuk informasi tentang memeriksa ukuran log transaksi, lihat [Memantau penggunaan ruang log](#) dalam [dokumentasi teknis SQL Server](#).

- Verifikasi bahwa rencana pemeliharaan Anda mengikuti praktik terbaik SQL server. Untuk informasi tentang praktik terbaik pemeliharaan server SQL, lihat [Strategi pemeliharaan indeks](#) dalam dokumentasi [teknis SQL Server](#).

Untuk memperbaiki masalah latensi selama pembuatan ulang indeks, coba yang berikut ini:

- Gunakan model BULK_LOGGED pemulihan untuk membangun kembali offline untuk mengurangi peristiwa yang harus diproses tugas.
- Jika memungkinkan, hentikan tugas selama pembangunan kembali indeks. Atau, cobalah untuk menjadwalkan pembangunan kembali indeks selama jam non-sibuk untuk mengurangi dampak lonjakan latensi.
- Cobalah untuk mengidentifikasi hambatan sumber daya yang memperlambat pembacaan DMS, seperti latensi disk atau throughput I/O, dan atasi.

Transaksi besar

Transaksi dengan banyak peristiwa, atau transaksi yang berjalan lama, menyebabkan log transaksi tumbuh. Hal ini menyebabkan pembacaan DMS memakan waktu lebih lama, menghasilkan latensi. Ini mirip dengan pembangunan kembali indeks efek pada kinerja replikasi.

Anda mungkin mengalami kesulitan mengidentifikasi masalah ini jika Anda tidak terbiasa dengan beban kerja tipikal pada database sumber. Untuk memecahkan masalah ini, lakukan hal berikut:

- Pertama, identifikasi waktu latensi melonjak menggunakan metrik ReadThroughput dan WriteThroughput CloudWatch metrik, atau dengan memeriksa pesan Pemantauan Throughput di log tugas.
- Periksa apakah ada kueri yang berjalan lama di database sumber selama lonjakan latensi. [Untuk informasi tentang kueri yang berjalan lama, lihat Memecahkan masalah kueri yang berjalan lambat di SQL Server dalam dokumentasi teknis SQL Server](#).
- Periksa apakah ukuran log transaksi aktif atau cadangan log telah meningkat. Untuk informasi selengkapnya, lihat [Memantau penggunaan ruang log](#) dalam [dokumentasi teknis SQL Server](#).

Untuk memperbaiki masalah ini, lakukan salah satu hal berikut:

- Perbaikan terbaik adalah merestrukturisasi transaksi Anda di sisi aplikasi sehingga mereka selesai dengan cepat.
- Jika Anda tidak dapat merestrukturisasi transaksi Anda, solusi jangka pendek adalah memeriksa kemacetan sumber daya seperti menunggu disk atau pertengkaran CPU. Jika Anda menemukan kemacetan di database sumber Anda, Anda dapat mengurangi latensi dengan meningkatkan disk, CPU, dan sumber daya memori untuk database sumber. Hal ini mengurangi pertentangan untuk sumber daya sistem, memungkinkan kueri DMS untuk menyelesaikan lebih cepat.

Interval polling MS-CDC yang salah dikonfigurasi untuk Amazon RDS SQL Server

Pengaturan interval polling yang salah dikonfigurasi pada instans Amazon RDS dapat menyebabkan log transaksi bertambah. Ini karena replikasi mencegah pemotongan log. Sementara tugas yang sedang berjalan mungkin terus bereplikasi dengan latensi minimal, menghentikan dan melanjutkan tugas, atau memulai tugas khusus CDC, dapat menyebabkan kegagalan tugas. Ini karena batas waktu saat memindai log transaksi besar.

Untuk memecahkan masalah interval polling yang salah konfigurasi, lakukan hal berikut:

- Periksa apakah ukuran log transaksi aktif meningkat, dan apakah penggunaan log mendekati 100 persen. Untuk informasi selengkapnya, lihat [Memantau penggunaan ruang log](#) dalam [dokumentasi teknis SQL Server](#).
- Periksa apakah pemotongan log tertunda dengan `alog_reuse_wait_desc` value. REPLICATION Untuk informasi selengkapnya, lihat [Log Transaksi \(SQL Server\)](#) di dokumentasi [teknis SQL Server](#).

Jika Anda menemukan masalah dengan salah satu item dalam daftar sebelumnya, atur interval polling MS-CDC. Untuk informasi tentang penyetelan interval pemungutan suara, lihat [Pengaturan yang disarankan saat menggunakan Amazon RDS for SQL Server sebagai sumber untuk AWS DMS](#)

Beberapa tugas CDC mereplikasi dari database sumber yang sama

Selama fase beban penuh, kami merekomendasikan pemisahan tabel di seluruh tugas untuk meningkatkan kinerja, untuk memisahkan tabel dependen secara logis, dan untuk mengurangi dampak kegagalan tugas. Namun, selama fase CDC, kami merekomendasikan konsolidasi tugas untuk meminimalkan pemindaian DMS. Selama fase CDC, setiap tugas DMS memindai log transaksi untuk peristiwa baru beberapa kali dalam satu menit. Karena setiap tugas berjalan secara independen, setiap tugas memindai setiap log transaksi satu per satu. Ini meningkatkan penggunaan disk dan CPU pada database SQL Server sumber. Akibatnya, sejumlah besar tugas yang berjalan

secara paralel dapat menyebabkan SQL Server menghambat pembacaan DMS, yang menyebabkan peningkatan latensi.

Anda mungkin mengalami kesulitan mengidentifikasi masalah ini jika banyak tugas dimulai secara bertahap. Gejala yang paling umum dari masalah ini adalah sebagian besar pemindaian tugas mulai memakan waktu lebih lama. Hal ini menyebabkan latensi yang lebih tinggi untuk pemindaian ini. SQL Server memprioritaskan beberapa pemindaian tugas, sehingga beberapa tugas menunjukkan latensi normal. Untuk memecahkan masalah ini, periksa `CDCLatencySource` metrik untuk semua tugas Anda. Jika beberapa tugas memiliki peningkatan `CDCLatencySource`, sementara beberapa tugas memiliki rendah `CDCLatencySource`, kemungkinan SQL Server membatasi pembacaan DMS Anda untuk beberapa tugas Anda.

Jika SQL Server membatasi pembacaan tugas Anda selama CDC, konsolidasikan tugas Anda untuk meminimalkan jumlah pemindaian DMS. Jumlah maksimum tugas yang dapat terhubung ke database sumber Anda tanpa membuat pertentangan tergantung pada faktor-faktor seperti kapasitas basis data sumber, tingkat pertumbuhan log transaksi, atau jumlah tabel. Untuk menentukan jumlah tugas yang ideal untuk skenario replikasi Anda, uji replikasi di lingkungan pengujian yang mirip dengan lingkungan produksi Anda.

Memecahkan masalah latensi target

Bagian ini berisi skenario yang dapat berkontribusi pada latensi target.

Topik

- [Masalah pengindeksan](#)
- [Pesan SORTER di log tugas](#)
- [Penguncian basis data](#)
- [Pencarian LOB lambat](#)
- [Multi-AZ, pencatatan audit dan pencadangan](#)

Masalah pengindeksan

Selama fase CDC, AWS DMS mereplikasi perubahan pada sumber dengan mengeksekusi pernyataan DHTML (menyisipkan, memperbarui, dan menghapus) pada target. Untuk migrasi heterogen menggunakan DMS, perbedaan optimasi indeks pada sumber dan target dapat menyebabkan penulisan ke target memakan waktu lebih lama. Ini menghasilkan latensi target dan masalah kinerja.

Untuk memecahkan masalah pengindeksan, lakukan hal berikut. Prosedur untuk langkah-langkah ini bervariasi untuk mesin database yang berbeda.

- Pantau waktu kueri untuk basis data target Anda. Membandingkan waktu eksekusi kueri pada target dan sumber dapat menunjukkan indeks mana yang memerlukan pengoptimalan.
- Aktifkan pencatatan untuk kueri yang berjalan lambat.

Untuk memperbaiki masalah pengindeksan untuk replikasi yang berjalan lama, lakukan hal berikut:

- Sesuaikan indeks pada basis data sumber dan target Anda sehingga waktu eksekusi kueri serupa pada sumber dan target.
- Bandingkan indeks sekunder yang digunakan dalam kueri DML untuk sumber dan target. Pastikan bahwa kinerja DHTML pada target sebanding dengan atau lebih baik daripada kinerja DMLnya sumber.

Perhatikan bahwa prosedur untuk mengoptimalkan indeks khusus untuk mesin database Anda. Tidak ada fitur DMS untuk menyetel sumber dan indeks target.

Pesan SORTER di log tugas

Jika titik akhir target tidak dapat mengikuti volume perubahan yang AWS DMS menuliskannya, tugas tersebut menyimpan perubahan pada instance replikasi. Jika cache tumbuh lebih besar dari ambang internal, tugas berhenti membaca perubahan lebih lanjut dari sumber. DMS melakukan ini untuk mencegah instance replikasi kehabisan penyimpanan, atau tugas macet saat membaca sejumlah besar peristiwa yang tertunda.

Untuk memecahkan masalah ini, periksa CloudWatch log untuk pesan yang mirip dengan salah satu dari berikut ini:

```
[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)
```

Jika log Anda berisi pesan yang mirip dengan pesan pertama, nonaktifkan pencatatan jejak apa pun untuk tugas tersebut, dan tingkatkan penyimpanan instance replikasi. Untuk informasi tentang meningkatkan penyimpanan instans replikasi, lihat [Mengubah instans replikasi](#).

Jika log Anda berisi pesan yang mirip dengan pesan kedua, lakukan hal berikut:

- Pindahkan tabel dengan banyak transaksi atau operasi DHTML yang berjalan lama ke tugas terpisah, jika mereka tidak memiliki dependensi pada tabel lain dalam tugas tersebut.
- Tingkatkan `MemoryLimitTotal` dan `MemoryKeepTime` pengaturan untuk menahan transaksi untuk durasi yang lebih lama dalam memori. Ini tidak akan membantu jika latensi dipertahankan, tetapi dapat membantu menjaga latensi turun selama ledakan singkat volume transaksional. Untuk informasi tentang pengaturan tugas ini, lihat [Mengubah pengaturan penyetelan pemrosesan](#).
- Evaluasi apakah Anda dapat menggunakan batch apply untuk transaksi Anda dengan menyetel `BatchApplyEnabled` ke `true`. Untuk informasi tentang `BatchApplyEnabled` pengaturan, lihat [Menargetkan pengaturan tugas metadata](#).

Penguncian basis data

Jika aplikasi mengakses database yang AWS DMS digunakan sebagai target replikasi, aplikasi dapat mengunci tabel yang DMS coba akses. Ini menciptakan pertenggaran kunci. Karena DMS menulis perubahan pada database target dalam urutan mereka terjadi pada sumber, penundaan untuk menulis ke satu tabel karena kontensi kunci membuat penundaan untuk menulis ke semua tabel.

Untuk memecahkan masalah ini, kueri database target untuk memeriksa apakah pertentangan kunci memblokir transaksi penulisan DMS. Jika database target memblokir transaksi tulis DMS, lakukan satu atau lebih hal berikut:

- Merestrukturisasi kueri Anda untuk melakukan perubahan lebih sering.
- Ubah pengaturan batas waktu kunci Anda.
- Partisi tabel Anda untuk meminimalkan kontensi kunci.

Perhatikan bahwa prosedur untuk mengoptimalkan kontensi kunci khusus untuk mesin database Anda. Tidak ada fitur DMS untuk menyetel kontensi kunci.

Pencarian LOB lambat

Saat AWS DMS mereplikasi kolom objek besar (LOB), ia melakukan pencarian pada sumber sebelum menulis perubahan pada target. Pencarian ini biasanya tidak menyebabkan latensi apa pun pada target, tetapi jika database sumber menunda pencarian karena penguncian, latensi target dapat melonjak.

Masalah ini biasanya sulit didiagnosis. Untuk memecahkan masalah ini, aktifkan debugging terperinci pada log tugas, dan bandingkan stempel waktu panggilan pencarian DMS LOB. Untuk informasi tentang mengaktifkan debugging mendetail, lihat. [Melihat dan mengelola AWSMencatat tugas DMS](#)

Untuk memperbaiki masalah ini, coba yang berikut ini:

- Tingkatkan kinerja kueri SELECT pada database sumber.
- Setel pengaturan DMS LOB. Untuk informasi tentang menyetel setelan LOB, lihat. [Melakukan migrasi objek biner large \(LOB\)](#)

Multi-AZ, pencatatan audit dan pencadangan

Untuk target Amazon RDS, latensi target dapat meningkat selama hal berikut:

- Cadangan
- Setelah mengaktifkan beberapa zona ketersediaan (Multi-AZ)
- Setelah mengaktifkan pencatatan database, seperti audit atau log kueri lambat.

Masalah-masalah ini biasanya sulit untuk didiagnosis. Untuk mengatasi masalah ini, pantau latensi untuk lonjakan berkala selama jendela pemeliharaan Amazon RDS atau periode pemuatan basis data yang berat.

Untuk memperbaiki masalah ini, coba yang berikut ini:

- Jika memungkinkan, selama migrasi jangka pendek, nonaktifkan Multi-AZ, backup, atau logging.
- Jadwalkan ulang jendela pemeliharaan Anda untuk periode aktivitas rendah.

Bekerja dengan skrip dukungan diagnostik di AWS DMS

Jika Anda mengalami masalah saat bekerja dengan AWS DMS, teknisi dukungan Anda mungkin memerlukan informasi lebih lanjut tentang basis data sumber atau target Anda. Kami ingin memastikan bahwa AWS Support mendapatkan sebanyak mungkin informasi yang diperlukan dalam waktu sesingkat mungkin. Oleh karena itu, kami mengembangkan skrip untuk kueri informasi ini untuk beberapa mesin basis data relasional utama.

Jika skrip dukungan tersedia untuk basis data, Anda dapat mengunduh skrip tersebut menggunakan tautan dalam topik skrip yang sesuai seperti dijelaskan berikut. Setelah memverifikasi dan meninjau skrip (dijelaskan berikut), Anda dapat menjalankannya sesuai dengan prosedur yang dijelaskan

dalam topik skrip. Ketika skrip yang dijalankan selesai, Anda dapat mengunggah outputnya ke kasus AWS Support (sekali lagi, dijelaskan berikut).

Sebelum menjalankan skrip, Anda dapat mendeteksi kesalahan yang mungkin timbul saat mengunduh atau menyimpan skrip dukungan. Untuk melakukan ini, bandingkan checksum untuk file skrip dengan nilai yang diberikan oleh AWS. AWS menggunakan algoritme SHA256 untuk checksum.

Untuk memverifikasi file skrip dukungan menggunakan checksum

1. Buka file checksum terbaru yang disediakan untuk memverifikasi skrip dukungan ini di <https://d2pwp9zz55emqw.cloudfront.net/sha256Check.txt>. Misalnya, file tersebut mungkin memiliki konten seperti berikut.

```
MYSQL dfafd0d511477c699f96c64693ad0b1547d47e74d5c5f2f2025b790b1422e3c8
ORACLE 6c41ebcfc99518cfa8a10cb2ce8943b153b2cc7049117183d0b5de3d551bc312
POSTGRES 6ccd274863d14f6f3146fbdabbba43f2d8d4c6a4c25380d7b41c71883aa4f9790
SQL_SERVER 971a6f2c46aec8d083d2b3b6549b1e9990af3a15fe4b922e319f4fdd358debe7
```

2. Jalankan perintah validasi SHA256 untuk sistem operasi Anda di direktori yang berisi file dukungan. Misalnya, pada sistem operasi macOS, Anda dapat menjalankan perintah berikut pada skrip dukungan Oracle yang akan dijelaskan dalam topik ini.

```
shasum -a 256 awsdms_support_collector_oracle.sql
```

3. Bandingkan hasil perintah dengan nilai yang ditunjukkan dalam file `sha256Check.txt` terbaru yang Anda buka. Kedua nilai harus cocok. Jika tidak cocok, hubungi teknisi dukungan Anda tentang ketidakcocokan tersebut dan bagaimana Anda dapat memperoleh file skrip dukungan bersih.

Jika Anda memiliki file skrip dukungan bersih, sebelum menjalankan skrip pastikan untuk membaca dan memahami SQL dari perspektif performa dan keamanan. Jika Anda merasa tidak nyaman untuk menjalankan salah satu SQL dalam skrip ini, Anda dapat mengomentari atau menghapus SQL yang bermasalah. Anda juga dapat berkonsultasi dengan teknisi dukungan Anda tentang solusi yang dapat diterima.

Setelah berhasil dan kecuali dinyatakan lain, skrip mengembalikan output dalam format HTML yang dapat dibaca. Skrip ini dirancang untuk mengecualikan data atau detail keamanan yang mungkin membahayakan bisnis Anda dari HTML ini. Skrip ini juga tidak melakukan modifikasi pada basis data Anda atau lingkungannya. Namun, jika Anda menemukan informasi dalam HTML yang membuat

Anda tidak nyaman jika dibagikan, jangan ragu untuk menghapus informasi yang bermasalah sebelum mengunggah HTML. Ketika HTML dapat diterima, unggah menggunakan Lampiran di Detail kasus dari kasus dukungan Anda.

Masing-masing topik berikut menjelaskan skrip yang tersedia untuk basis data AWS DMS yang didukung dan cara menjalankannya. Teknisi dukungan Anda akan mengarahkan Anda ke skrip tertentu yang didokumentasikan berikut.

Topik

- [Skrip dukungan diagnostik Oracle](#)
- [Skrip dukungan diagnostik SQL Server](#)
- [Skrip dukungan diagnostik untuk basis data yang kompatibel dengan MySQL](#)
- [Skrip dukungan diagnostik PostgreSQL](#)

Skrip dukungan diagnostik Oracle

Berikut ini, Anda dapat menemukan skrip dukungan diagnostik yang tersedia untuk menganalisis basis data on premise atau Amazon RDS for Oracle Database di konfigurasi migrasi AWS DMS. Skrip ini bekerja dengan titik akhir sumber atau target. Semua skrip ditulis untuk berjalan di utilitas baris perintah SQL* Plus. Untuk informasi lebih lanjut tentang penggunaan utilitas ini, lihat [Menggunakan Baris Perintah SQL](#) dalam dokumentasi Oracle.

Sebelum menjalankan skrip, pastikan bahwa akun pengguna yang Anda gunakan memiliki izin yang diperlukan untuk mengakses basis data Oracle Anda. Pengaturan izin yang ditampilkan menganggap pengguna membuat sebagai berikut.

```
CREATE USER script_user IDENTIFIED BY password;
```

Untuk basis data on premise, atur izin minimum seperti yang ditunjukkan berikut untuk *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;

```

Untuk basis data Amazon RDS, atur izin minimum seperti yang ditunjukkan berikut.

```

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');
```

Berikut ini, Anda dapat menemukan deskripsi cara mengunduh, meninjau, dan menjalankan setiap skrip dukungan SQL* Plus yang tersedia untuk Oracle. Anda juga dapat menemukan cara meninjau dan mengunggah output ke kasus AWS Support.

Topik

- [skrip awsdms_support_collector_oracle.sql](#)

skrip awsdms_support_collector_oracle.sql

Unduh skrip [awsdms_support_collector_oracle.sql](#).

Skrip ini mengumpulkan informasi tentang konfigurasi basis data Oracle Anda. Ingatlah untuk memverifikasi checksum pada skrip, dan jika checksum memverifikasi, tinjau kode SQL di dalam

skrip untuk mengomentari salah satu kode yang tidak nyaman Anda jalankan. Setelah Anda puas dengan integritas dan isi skrip, Anda bisa menjalankannya.

Untuk menjalankan skrip dan mengunggah hasilnya ke kasus dukungan Anda

1. Jalankan skrip dari lingkungan basis data Anda menggunakan baris perintah SQL* Plus berikut.

```
SQL> @awsdms_support_collector_oracle.sql
```

<result>

Skrip menampilkan deskripsi singkat dan prompt untuk melanjutkan atau membatalkan proses menjalankan skrip. Tekan [Enter] untuk melanjutkan.

</result>

2. Pada prompt berikut, masukkan nama salah satu skema yang ingin Anda migrasikan.
3. Pada prompt berikut, masukkan nama pengguna (*script_user*) yang telah Anda tetapkan untuk terhubung ke basis data.
4. Pada prompt berikut, masukkan jumlah hari data yang ingin Anda periksa, atau terima jumlah default. Skrip kemudian mengumpulkan data yang ditentukan dari basis data Anda.

<result>

Setelah skrip selesai, skrip akan menampilkan nama file HTML output, misalnya `dms_support_oracle-2020-06-22-13-20-39-ORCL.html`. Skrip menyimpan file ini di direktori kerja Anda.

</result>

5. Tinjau file HTML ini dan hapus informasi yang membuat Anda tidak nyaman jika dibagikan. Ketika Anda merasa HTML dapat dibagikan, unggah file ke kasus AWS Support. Untuk informasi lebih lanjut tentang mengunggah file ini, lihat [Bekerja dengan skrip dukungan diagnostik di AWS DMS](#).

Skrip dukungan diagnostik SQL Server

Berikut ini, Anda dapat menemukan deskripsi skrip dukungan diagnostik yang tersedia untuk menganalisis basis data on premise atau Amazon RDS for SQL Server di konfigurasi migrasi AWS DMS. Skrip ini bekerja dengan titik akhir sumber atau target. Untuk basis data on premise, jalankan skrip ini di utilitas baris perintah sqlcmd. Untuk informasi lebih lanjut tentang penggunaan utilitas ini, lihat [sqlcmd - Gunakan utilitas](#) dalam dokumentasi Microsoft.

Untuk basis data Amazon RDS, Anda tidak dapat terhubung menggunakan utilitas baris perintah sqlcmd. Sebaliknya, jalankan skrip ini menggunakan alat klien yang terhubung ke Amazon RDS SQL Server.

Sebelum menjalankan skrip, pastikan bahwa akun pengguna yang Anda gunakan memiliki izin yang diperlukan untuk mengakses basis data SQL Server. Untuk basis data on premise dan Amazon RDS, Anda dapat menggunakan izin yang sama dengan yang Anda gunakan untuk mengakses database SQL Server tanpa peran SysAdmin.

Topik

- [Menyiapkan izin minimum untuk database SQL Server lokal](#)
- [Menyiapkan izin minimum untuk database Amazon RDS SQL Server](#)
- [Menyiapkan replikasi yang sedang berlangsung pada SQL Server mandiri: Tanpa peran sysadmin](#)
- [Menyiapkan replikasi yang sedang berlangsung di SQL Server di lingkungan grup ketersediaan: Tanpa peran sysadmin](#)
- [Skrip Dukungan SQL Server](#)

Menyiapkan izin minimum untuk database SQL Server lokal

Untuk mengatur izin minimum untuk menjalankan basis data SQL Server on premise

1. Buat akun SQL Server baru dengan autentikasi kata sandi menggunakan SQL Server Management Studio (SSMS), misalnya *on-prem-user*.
2. Di bagian Pemetaan pengguna dari SSMS, pilih basis data MSDB dan MASTER (yang memberikan izin publik), dan tetapkan peran DB_OWNER untuk basis data tempat Anda ingin menjalankan skrip.
3. Buka menu konteks (klik kanan) untuk akun baru, lalu pilih Keamanan untuk secara eksplisit memberikan hak istimewa Connect SQL.
4. Jalankan perintah pemberian berikut.

```
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;
```

Menyiapkan izin minimum untuk database Amazon RDS SQL Server

Untuk menjalankan dengan izin minimum untuk basis data Amazon RDS SQL Server

1. Buat akun SQL Server baru dengan autentikasi kata sandi menggunakan SQL Server Management Studio (SSMS), misalnya *rds-user*.
2. Di bagian Pemetaan Pengguna dari SSMS, pilih basis data MSDB (yang memberikan izin publik), dan tetapkan peran DB_OWNER untuk basis data tempat Anda ingin menjalankan skrip.
3. Buka menu konteks (klik kanan) untuk akun baru, lalu pilih Keamanan untuk secara eksplisit memberikan hak istimewa Connect SQL.
4. Jalankan perintah pemberian berikut.

```
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;
```

Menyiapkan replikasi yang sedang berlangsung pada SQL Server mandiri: Tanpa peran sysadmin

Bagian ini menjelaskan cara mengatur replikasi berkelanjutan untuk sumber database SQL Server mandiri yang tidak memerlukan akun pengguna untuk memiliki hak istimewa sysadmin.

Note

Setelah menjalankan langkah-langkah di bagian ini, pengguna DMS non-sysadmin akan memiliki izin untuk melakukan hal berikut:

- Baca perubahan dari file log transaksi online
- Akses disk untuk membaca perubahan dari file cadangan log transaksional
- Menambahkan atau mengubah publikasi yang digunakan DMS
- Tambahkan artikel ke publikasi

1. Siapkan Microsoft SQL Server untuk Replikasi seperti yang dijelaskan dalam [Menangkap perubahan data untuk SQL Server yang dikelola sendiri di tempat atau di Amazon EC2](#)

2. Aktifkan MS-REPLIKASI pada database sumber. Ini dapat dilakukan secara manual atau dengan menjalankan tugas sekali sebagai pengguna sysadmin.
3. Buat awsdms skema pada database sumber menggunakan skrip berikut:

```
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

@dlim 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 @dml_pos integer;  
  
declare @dml_len integer;  
  
set @dml_len = len(@dml);  
  
while (charindex(@dml,@plist)>0)  
  
begin  
  
set @dml_pos = charindex(@dml,@plist);  
  
set @partition_id = cast( ltrim(rtrim(substring(@plist,1,@dml_pos-1))) as bigint);  
  
insert into @partitionsTable (pid) values (@partition_id)  
  
set @plist = substring(@plist,@dml_pos+@dml_len,len(@plist));  
  
end  
  
set @partition_id = cast (ltrim(rtrim(@plist)) as bigint);  
  
insert into @partitionsTable (pid) values ( @partition_id );  
  
return  
  
end  
  
GO
```

4. Buat [awsdms].[rtm_dump_dblog] prosedur pada database Master menggunakan skrip berikut:

```
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)

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.awsmdms.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)  
  
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. Buat sertifikat pada database Master menggunakan skrip berikut:

```
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. Buat login dari sertifikat menggunakan skrip berikut:

```
Use [master]
Go

CREATE LOGIN awsdms_rtm_dump_dblog_login FROM CERTIFICATE
[awsdms_rtm_dump_dblog_cert];
```

7. Tambahkan login ke peran server sysadmin menggunakan skrip berikut:

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_dump_dblog_login];
```

8. Tambahkan tanda tangan ke [master].[awsdms].[rtm_dump_dblog] menggunakan sertifikat, menggunakan skrip berikut:

```
Use [master]
GO
ADD SIGNATURE
TO [master].[awsdms].[rtm_dump_dblog] BY CERTIFICATE [awsdms_rtm_dump_dblog_cert]
WITH PASSWORD = '@5trongpassword';
```

Note

Jika Anda membuat ulang prosedur yang disimpan, Anda perlu menambahkan tanda tangan lagi.

9. Buat [awsdms].[rtm_position_1st_timestamp] pada database Master menggunakan skrip berikut:

```
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)
```



```
WITH SUBJECT = N'Certificate for FN_POSITION_1st_TIMESTAMP Permissions';
```

11. Buat login dari sertifikat menggunakan skrip berikut:

```
Use [master]
Go
CREATE LOGIN awsdms_rtm_position_1st_timestamp_login FROM CERTIFICATE
[awsdms_rtm_position_1st_timestamp_cert];
```

12. Tambahkan login ke peran sysadmin menggunakan skrip berikut:

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_position_1st_timestamp_login];
```

13. Tambahkan tanda tangan ke [master].[awsdms].[rtm_position_1st_timestamp] menggunakan sertifikat, menggunakan skrip berikut:

```
Use [master]
GO
ADD SIGNATURE
TO [master].[awsdms].[rtm_position_1st_timestamp]
BY CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
WITH PASSWORD = '@5trongpassword';
```

14. Berikan pengguna DMS mengeksekusi akses ke prosedur tersimpan baru menggunakan skrip berikut:

```
use master
go
GRANT execute on [awsdms].[rtm_position_1st_timestamp] to dms_user;
```

15. Buat pengguna dengan izin dan peran berikut di masing-masing database berikut:

 Note

Anda harus membuat akun pengguna dmsnosysadmin dengan SID yang sama pada setiap replika. Kueri SQL berikut dapat membantu memverifikasi nilai SID akun dmsnosysadmin pada setiap replika. Untuk informasi selengkapnya tentang membuat pengguna, lihat [MEMBUAT PENGGUNA \(Transact-SQL\) di dokumentasi server Microsoft SQL](#). [Untuk informasi selengkapnya tentang membuat akun pengguna SQL untuk database Azure SQL, lihat Replikasi geo aktif.](#)

```
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]
```

Jalankan skrip berikut pada database sumber:

```
EXEC sp_addrolemember N'db_owner', N'DMS_user'
use Source_DB
go
```

16. Terakhir, tambahkan Extra Connection Attribute (ECA) ke titik akhir SQL Server sumber:

```
enableNonSysadminWrapper=true;
```

Menyiapkan replikasi yang sedang berlangsung di SQL Server di lingkungan grup ketersediaan: Tanpa peran sysadmin

Bagian ini menjelaskan cara mengatur replikasi berkelanjutan untuk sumber database SQL Server di lingkungan grup ketersediaan yang tidak memerlukan akun pengguna untuk memiliki hak istimewa sysadmin.

Note

Setelah menjalankan langkah-langkah di bagian ini, pengguna DMS non-sysadmin akan memiliki izin untuk melakukan hal berikut:

- Baca perubahan dari file log transaksi online
- Akses disk untuk membaca perubahan dari file cadangan log transaksional
- Menambahkan atau mengubah publikasi yang digunakan DMS
- Tambahkan artikel ke publikasi

Untuk mengatur replikasi yang sedang berlangsung tanpa menggunakan pengguna sysadmin di lingkungan Grup Ketersediaan

1. Siapkan Microsoft SQL Server untuk Replikasi seperti yang dijelaskan dalam [Menangkap perubahan data untuk SQL Server yang dikelola sendiri di tempat atau di Amazon EC2](#)
2. Aktifkan MS-REPLIKASI pada database sumber. Ini dapat dilakukan secara manual atau dengan menjalankan tugas sekali menggunakan pengguna sysadmin.

Note

Anda harus mengkonfigurasi distributor MS-REPLICATION sebagai lokal atau dengan cara yang memungkinkan akses ke pengguna non-sysadmin melalui server tertaut terkait.

3. Jika opsi Exclusive use sp_repldone dalam satu titik akhir tugas diaktifkan, hentikan pekerjaan MS-REPLICATION Log Reader.
4. Lakukan langkah-langkah berikut pada setiap replika:
 1. Buat skema [awsdms] [awsdms] di database master:

```
CREATE SCHEMA [awsdms]
```

2. Buat fungsi [awsdms].[split_partition_list] tabel bernilai pada database 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. Buat [awsdms].[rtm_dump_dblog] prosedur pada database 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)
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)
    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. Buat sertifikat pada Master Database:

```
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. Buat login dari sertifikat:

```
USE [master]
GO
CREATE LOGIN awsdms_rtm_dump_dblog_login FROM CERTIFICATE
    [awsdms_rtm_dump_dblog_cert];
```

6. Tambahkan login ke peran server sysadmin:

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_dump_dblog_login];
```

7. Tambahkan tanda tangan ke [master].[awsdms].[rtm_dump_dblog] menggunakan sertifikat:

```
USE [master]
GO

ADD SIGNATURE
    TO [master].[awsdms].[rtm_dump_dblog]
    BY CERTIFICATE [awsdms_rtm_dump_dblog_cert]
    WITH PASSWORD = '@hardpassword1';
```

Note

Jika Anda membuat ulang prosedur yang disimpan, Anda perlu menambahkan tanda tangan lagi.

8. Buat [awsdms].[rtm_position_1st_timestamp] prosedur pada database 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, default,''+@nl+
        @tb+'default, default, default, default, default, default, default, default,''+@nl+

```

```

@tb+'default, default, default, default, default, default, default, default,'+@n1+
@tb+'default, default, default, default, default, default, default, default,'+@n1+
@tb+'default, default, default, default, default, default, default, default,'+@n1+
@tb+'default, default, default, default, default, default, default, default)+'@n1+
'WHERE operation=''LOP_BEGIN_XACT''' +@n1+
'AND [Begin Time]>= cast('+'''+@1stTimeStamp+'''+ ' as datetime)+'@n1

--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. Buat sertifikat pada database 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. Buat login dari sertifikat:

```

USE [master]
GO
CREATE LOGIN awsdms_rtm_position_1st_timestamp_login FROM CERTIFICATE
    [awsdms_rtm_position_1st_timestamp_cert];

```

11. Tambahkan login ke peran server sysadmin:

```

ALTER SERVER ROLE [sysadmin] ADD MEMBER
    [awsdms_rtm_position_1st_timestamp_login];

```

12. Tambahkan tanda tangan ke [master].[awsdms].[rtm_position_1st_timestamp] prosedur menggunakan sertifikat:

```

USE [master]
GO

```

```
ADD SIGNATURE
  TO [master].[awsdms].[rtm_position_1st_timestamp]
  BY CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
  WITH PASSWORD = '@hardpassword1';
```

Note

Jika Anda membuat ulang prosedur yang disimpan, Anda perlu menambahkan tanda tangan lagi.

13Buat pengguna dengan izin/peran berikut di setiap database berikut:

Note

Anda harus membuat akun pengguna dmsnosysadmin dengan SID yang sama pada setiap replika. Kueri SQL berikut dapat membantu memverifikasi nilai SID akun dmsnosysadmin pada setiap replika. Untuk informasi selengkapnya tentang membuat pengguna, lihat [MEMBUAT PENGGUNA \(Transact-SQL\) di dokumentasi server Microsoft SQL](#). [Untuk informasi selengkapnya tentang membuat akun pengguna SQL untuk database Azure SQL, lihat Replikasi geo aktif.](#)

```
SELECT @@servername servername, name, sid, create_date, modify_date
  FROM sys.server_principals
  WHERE name = 'dmsnosysadmin';
```

14Berikan izin pada database master pada setiap replika:

```
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. Berikan izin pada database msdb pada setiap replika:

```
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. Tambahkan db_owner peran dmsnosysadmin ke database sumber. Karena database disinkronkan, Anda dapat menambahkan peran pada replika utama saja.

```
use <source DB>  
GO  
EXEC sp_addrolemember N'db_owner', N'dmsnosysadmin'
```

Skrip Dukungan SQL Server

Topik berikut menjelaskan cara mengunduh, meninjau, dan menjalankan setiap skrip dukungan yang tersedia untuk SQL Server. Topik berikut juga menjelaskan cara meninjau dan mengunggah output skrip ke kasus AWS Support Anda.

Topik

- [skrip awsdms_support_collector_sql_server.sql](#)

skrip awsdms_support_collector_sql_server.sql

Unduh skrip [awsdms_support_collector_sql_server.sql](#).

Note

Jalankan skrip dukungan diagnostik SQL Server ini pada SQL Server 2014 dan versi yang lebih tinggi saja.

Skrip ini mengumpulkan informasi tentang konfigurasi basis data SQL Server Anda. Ingatlah untuk memverifikasi checksum pada skrip, dan jika checksum memverifikasi, tinjau kode SQL di dalam

skrip untuk mengomentari salah satu kode yang tidak nyaman Anda jalankan. Setelah Anda puas dengan integritas dan isi skrip, Anda bisa menjalankannya.

Untuk menjalankan skrip untuk basis data SQL Server on premise

1. Jalankan skrip menggunakan baris perintah sqlcmd berikut.

```
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
```

Parameter perintah sqlcmd yang ditentukan mencakup hal-hal berikut:

- -U – Nama pengguna basis data.
 - -P – Kata sandi pengguna basis data.
 - -S – Nama server basis data SQL Server.
 - -y – Lebar maksimum output kolom dari utilitas sqlcmd. Nilai 0 menentukan kolom dengan lebar tak terbatas.
 - -i – Jalur skrip dukungan dijalankan, dalam hal ini `awsdms_support_collector_sql_server.sql`.
 - -o – Jalur file HTML output, dengan nama file yang Anda tentukan, berisi informasi konfigurasi basis data yang dikumpulkan.
 - -d – Nama basis data SQL Server.
2. Setelah skrip selesai, tinjau file HTML output dan hapus informasi yang membuat Anda tidak nyaman jika dibagikan. Ketika Anda merasa HTML dapat dibagikan, unggah file ke kasus AWS Support. Untuk informasi lebih lanjut tentang mengunggah file ini, lihat [Bekerja dengan skrip dukungan diagnostik di AWS DMS](#).

Dengan Amazon RDS for SQL Server, Anda tidak dapat terhubung menggunakan utilitas baris perintah sqlcmd, jadi gunakan prosedur berikut.

Untuk menjalankan skrip untuk basis data RDS SQL Server

1. Jalankan skrip menggunakan alat klien yang memungkinkan Anda untuk terhubung ke RDS SQL Server sebagai pengguna Master dan simpan output sebagai file HTML.

2. Tinjau file HTML output dan hapus informasi yang membuat Anda tidak nyaman jika dibagikan. Ketika Anda merasa HTML dapat dibagikan, unggah file ke kasus AWS Support. Untuk informasi lebih lanjut tentang mengunggah file ini, lihat [Bekerja dengan skrip dukungan diagnostik di AWS DMS](#).

Skrip dukungan diagnostik untuk basis data yang kompatibel dengan MySQL

Berikut ini, Anda dapat menemukan skrip dukungan diagnostik yang tersedia untuk menganalisis basis data on premise atau basis data yang kompatibel dengan Amazon RDS for MySQL di konfigurasi migrasi AWS DMS Anda. Skrip ini bekerja dengan titik akhir sumber atau target. Semua skrip ditulis untuk berjalan pada baris perintah MySQL SQL.

Untuk informasi tentang menginstal klien MySQL, lihat [Menginstal MySQL Shell di dokumentasi MySQL](#). Untuk informasi tentang menggunakan klien MySQL, lihat [Menggunakan Perintah MySQL Shell dalam dokumentasi MySQL](#).

Sebelum menjalankan skrip, pastikan bahwa akun pengguna yang Anda gunakan memiliki izin yang diperlukan untuk mengakses basis data yang kompatibel dengan MySQL Anda. Gunakan prosedur berikut untuk membuat akun pengguna dan memberikan izin minimum yang diperlukan untuk menjalankan skrip ini.

Untuk mengatur akun pengguna dengan izin minimum untuk menjalankan skrip ini

1. Buat pengguna untuk menjalankan skrip.

```
create user 'username'@'hostname' identified by password;
```

2. Berikan perintah select pada basis data untuk menganalisis mereka.

```
grant select on database-name.* to username;  
grant replication client on *.* to username;
```

3.

```
grant execute on procedure mysql.rds_show_configuration to username;
```

Topik berikut menjelaskan cara mengunduh, meninjau, dan menjalankan setiap skrip dukungan yang tersedia untuk basis data yang kompatibel dengan MySQL. Topik berikut juga menjelaskan cara meninjau dan mengunggah output skrip ke kasus AWS Support Anda.

Topik

- [skrip awsdms_support_collector_MySQL.sql](#)

skrip awsdms_support_collector_MySQL.sql

Unduh skrip [awsdms_support_collector_MySQL.sql](#).

Skrip ini mengumpulkan informasi tentang konfigurasi basis data yang kompatibel dengan MySQL Anda. Ingatlah untuk memverifikasi checksum pada skrip, dan jika checksum memverifikasi, tinjau kode SQL di dalam skrip untuk mengomentari salah satu kode yang tidak nyaman Anda jalankan. Setelah Anda puas dengan integritas dan isi skrip, Anda bisa menjalankannya.

Jalankan skrip setelah terhubung ke lingkungan basis data Anda menggunakan baris perintah.

Untuk menjalankan skrip ini dan mengunggah hasilnya ke kasus dukungan Anda

1. Hubungkan ke basis data Anda dengan menggunakan perintah `mysql` berikut.

```
mysql -h hostname -P port -u username database-name
```

2. Jalankan skrip menggunakan perintah `source` `mysql` berikut.

```
mysql> source awsdms_support_collector_MySQL_compatible_DB.sql
```

Tinjau laporan yang dihasilkan dan hapus informasi yang membuat Anda tidak nyaman jika dibagikan. Ketika Anda merasa konten dapat dibagikan, unggah file ke kasus AWS Support Anda. Untuk informasi lebih lanjut tentang mengunggah file ini, lihat [Bekerja dengan skrip dukungan diagnostik di AWS DMS](#).

Note

- Jika Anda sudah memiliki akun pengguna dengan hak istimewa yang diperlukan yang dijelaskan di [Skrip dukungan diagnostik untuk basis data yang kompatibel dengan MySQL](#), Anda juga dapat menggunakan akun pengguna yang sudah ada untuk menjalankan skrip.

- Ingatlah untuk terhubung ke basis data Anda sebelum menjalankan skrip.
- Skrip menghasilkan output dalam format teks.
- Mengingat praktik keamanan terbaik, jika Anda membuat akun pengguna baru hanya untuk mengeksekusi skrip dukungan diagnostik MySQL ini, kami sarankan Anda untuk menghapus akun pengguna ini setelah skrip berhasil dieksekusi.

Skrip dukungan diagnostik PostgreSQL

Berikut ini, Anda dapat menemukan skrip dukungan diagnostik yang tersedia untuk menganalisis PostgreSQL RDBMS (on premise, Amazon RDS, atau Aurora PostgreSQL) di konfigurasi migrasi AWS DMS. Skrip ini bekerja dengan titik akhir sumber atau target. Semua skrip ditulis untuk berjalan di utilitas baris perintah psql.

Sebelum menjalankan skrip ini, pastikan bahwa akun pengguna yang Anda gunakan memiliki izin yang diperlukan berikut ini untuk mengakses PostgreSQL RDBMS:

- PostgreSQL 10.x atau lebih tinggi - Akun pengguna dengan izin eksekusi pada fungsi tersebut.
`pg_catalog.pg_ls_waldir`
- PostgreSQL 9.x atau sebelumnya – Sebuah akun pengguna dengan izin default.

Sebaiknya gunakan akun yang sudah ada dengan izin yang sesuai untuk menjalankan skrip ini.

Jika Anda perlu membuat akun pengguna baru atau memberikan izin ke akun yang sudah ada untuk menjalankan skrip ini, Anda dapat menjalankan perintah SQL berikut untuk PostgreSQL RDBMS berdasarkan versi PostgreSQL.

Untuk memberikan izin akun untuk menjalankan skrip ini untuk database PostgreSQL versi 10.x atau lebih tinggi

- Lakukan salah satu dari berikut:
 - Untuk akun pengguna baru, jalankan berikut ini.

```
CREATE USER script_user WITH PASSWORD 'password';  
GRANT EXECUTE ON FUNCTION pg_catalog.pg_ls_waldir TO script_user;
```

- Untuk akun pengguna yang sudah ada, jalankan berikut ini.

```
GRANT EXECUTE ON FUNCTION pg_catalog.pg_ls_waldir TO script_user;
```

Untuk memberikan izin akun guna menjalankan skrip ini untuk basis data PostgreSQL 9.x atau sebelumnya

- Lakukan salah satu dari berikut ini:
 - Untuk akun pengguna baru, jalankan berikut ini dengan izin default.

```
CREATE USER script_user WITH PASSWORD password;
```

- Untuk akun pengguna yang sudah ada, gunakan izin yang sudah ada.

Note

Skrip ini tidak mendukung fungsionalitas tertentu yang terkait dengan menemukan ukuran WAL untuk basis data PostgreSQL 9.x dan sebelumnya. Untuk informasi lebih lanjut, bekerjalah dengan AWS Support.

Topik berikut menjelaskan cara mengunduh, meninjau, dan menjalankan setiap skrip dukungan yang tersedia untuk PostgreSQL. Topik berikut juga menjelaskan cara meninjau dan mengunggah output skrip ke kasus AWS Support Anda.

Topik

- [skrip awsdms_support_collector_postgres.sql](#)

skrip awsdms_support_collector_postgres.sql

Unduh skrip [awsdms_support_collector_postgres.sql](#).

Skrip ini mengumpulkan informasi tentang konfigurasi basis data PostgreSQL Anda. Ingatlah untuk memverifikasi checksum pada skrip. Jika checksum memverifikasi, tinjau kode SQL dalam skrip untuk mengomentari salah satu kode yang tidak nyaman Anda jalankan. Setelah Anda puas dengan integritas dan isi skrip, Anda bisa menjalankannya.

Note

Anda dapat menjalankan skrip ini dengan psql client versi 10 atau lebih tinggi.

Anda dapat menggunakan prosedur berikut untuk menjalankan skrip ini baik dari lingkungan basis data Anda atau dari baris perintah. Dalam kedua kasus, Anda kemudian dapat mengunggah file Anda ke AWS Support.

Untuk menjalankan skrip ini dan mengunggah hasilnya ke kasus dukungan Anda

1. Lakukan salah satu dari berikut ini:

- Jalankan skrip dari lingkungan basis data Anda menggunakan baris perintah psql berikut.

```
dbname=# \i awsdms_support_collector_postgres.sql
```

Pada prompt berikut, masukkan nama salah satu skema yang ingin Anda migrasikan.

Pada prompt berikut, masukkan nama pengguna (*script_user*) yang telah Anda tetapkan untuk terhubung ke basis data.

- Jalankan skrip berikut langsung dari baris perintah. Opsi ini menghindari prompt sebelum eksekusi skrip.

```
psql -h database-hostname -p port -U script_user -d database-name -f  
awsdms_support_collector_postgres.sql
```

2. Tinjau file HTML output dan hapus informasi yang membuat Anda tidak nyaman jika dibagikan. Ketika Anda merasa HTML dapat dibagikan, unggah file ke kasus AWS Support. Untuk informasi lebih lanjut tentang mengunggah file ini, lihat [Bekerja dengan skrip dukungan diagnostik di AWS DMS](#).

Bekerja dengan dukungan AWS DMS diagnostik AMI

Jika Anda mengalami masalah terkait jaringan saat bekerja dengan AWS DMS, teknisi dukungan Anda mungkin memerlukan informasi lebih lanjut tentang konfigurasi jaringan Anda. Kami ingin memastikan bahwa AWS Support mendapatkan sebanyak mungkin informasi yang diperlukan dalam

waktu sesingkat mungkin. Oleh karena itu, kami mengembangkan Amazon EC2 AMI bawaan dengan alat diagnostik untuk AWS DMS menguji lingkungan jaringan Anda.

Tes diagnostik yang diinstal pada gambar mesin Amazon (AMI) meliputi yang berikut:

- Virtual Private Cloud (VPC)
- Kehilangan paket jaringan
- Latensi jaringan
- Ukuran Unit Transmisi Maksimum (MTU)

Topik

- [Luncurkan instans Amazon EC2 AWS DMS diagnostik baru](#)
- [Membuat IAM role](#)
- [Jalankan Tes Diagnostik](#)
- [Langkah Selanjutnya](#)
- [ID AMI menurut wilayah](#)

 Note

Jika Anda mengalami masalah kinerja dengan sumber Oracle Anda, Anda dapat mengevaluasi kinerja baca redo Oracle atau log arsip Anda untuk menemukan cara untuk meningkatkan kinerja. Untuk informasi selengkapnya, lihat [Mengevaluasi kinerja baca pengulangan Oracle atau log arsip](#).

Luncurkan instans Amazon EC2 AWS DMS diagnostik baru

Di bagian ini, Anda meluncurkan instans Amazon EC2 baru. Untuk informasi tentang cara meluncurkan instans Amazon EC2, lihat [Memulai tutorial instans Amazon EC2 Linux](#) di Panduan Pengguna Amazon [EC2](#).

Luncurkan instans Amazon EC2 dengan pengaturan berikut:

- Untuk Gambar Aplikasi dan OS (Gambar Mesin Amazon), cari AMI DMS-DIAG-AMI. Jika Anda masuk ke konsol, Anda dapat mencari AMI dengan [kueri ini](#) Untuk ID AMI AMI AMI AWS Diagnostik di wilayah Anda, lihat [ID AMI menurut wilayah](#) berikut.

- Untuk jenis Instance, kami sarankan Anda memilih t2.micro.
- Untuk Pengaturan Jaringan, pilih VPC yang sama dengan yang digunakan instans replikasi Anda.

Setelah instance aktif, sambungkan ke instance. Untuk informasi tentang menghubungkan ke instans Amazon EC2 Linux, lihat [Connect ke instans Linux Anda](#).

Membuat IAM role

Jika Anda ingin menjalankan tes diagnostik pada instance replikasi menggunakan izin minimum yang diperlukan, buat peran IAM yang menggunakan kebijakan izin berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "dms:DescribeEndpoints",
        "dms:DescribeTableStatistics",
        "dms:DescribeReplicationInstances",
        "dms:DescribeReplicationTasks",
        "secretsmanager:GetSecretValue"
      ],
      "Resource": "*"
    }
  ]
}
```

Lampirkan peran ke pengguna IAM baru. Untuk informasi tentang membuat peran, kebijakan, dan pengguna IAM, lihat bagian berikut di [Panduan Pengguna IAM](#):

- [Memulai dengan IAM](#)
- [Membuat Peran IAM](#)
- [Membuat Kebijakan IAM](#)

Perintah ini menampilkan data diagnostik tentang VPC instans replikasi Anda, transmisi paket jaringan, latensi jaringan, dan ukuran Unit Transmisi Maksimum (MTU) jaringan.

```

#####
#
#
#   AWS DMS Diagnostic
#   Date: 07-13-2022
#
#
#   aws region: us-east-2
#
#
#####
==== DMS DIAG Info ====
Public IP: 3.22.100.10
Private IP: 172.30.0.240
Instance ID: i-04829b2beb8214602
Instance MAC: 02:58:04:b5:52:28
Instance Type: t2.micro
Instance Sec Group: DMS-EC2-sec-group
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 (dms-04829b2beb8214602-postgres-dev-instance-1.cucdvzaur7nk.us-east-2.rds.amazonaws.com:5432)
>>>>Result: 10 packets transmitted, 10 packets received, 0% packet loss
    Looks good with no issue. <<<<<

3.) Check Target endpoint (rds-postgres-instance-1.cucdvzaur7nk.us-east-2.rds.amazonaws.com:5432)
>>>>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 (dms-04829b2beb8214602-postgres-dev-instance-1.cucdvzaur7nk.us-east-2.rds.amazonaws.com:5432)
>>>>Result: round-trip min/avg/max = 1.0/1.1/1.2 ms
    Looks good with no issue. <<<<<

3.) Check Target endpoint (rds-postgres-instance-1.cucdvzaur7nk.us-east-2.rds.amazonaws.com:5432)
>>>>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 (dms-04829b2beb8214602-postgres-dev-instance-1.cucdvzaur7nk.us-east-2.rds.amazonaws.com:5432)
>>>>Result: MTU setting looks good. Local MTU (9001) matches remote MTU (9001) <<<<<

3.) Check Target endpoint (rds-postgres-instance-1.cucdvzaur7nk.us-east-2.rds.amazonaws.com:5432)
>>>>Result: MTU setting looks good. Local MTU (9001) matches remote MTU (9001) <<<<<

==== End network MTU check ====

```

Perform AMI Diag EC2 VPC Check**Perform Network Packet Test****Returns Test Results and Recommendation****Perform Network Latency Test****Perform Network Maximum Transmission Unit (MTU) Check**

Langkah Selanjutnya

Bagian berikut menjelaskan informasi pemecahan masalah berdasarkan hasil tes diagnostik jaringan:

Tes VPC

Pengujian ini memverifikasi bahwa instans Amazon EC2 diagnostik berada dalam VPC yang sama dengan instance replikasi. Jika instans Amazon EC2 diagnostik tidak berada dalam VPC yang sama dengan instance replikasi Anda, hentikan dan buat lagi di VPC yang benar. Anda tidak dapat mengubah VPC instans Amazon EC2 setelah Anda membuatnya.

Tes kehilangan paket jaringan

Tes ini mengirimkan 10 paket ke titik akhir berikut dan memeriksa kehilangan paket:

- Layanan metadata AWS DMS Amazon EC2 pada port 80
- Titik akhir sumber
- Titik akhir target

Semua paket harus tiba dengan sukses. Jika ada paket yang hilang, konsultasikan dengan insinyur jaringan untuk menentukan masalah dan menemukan solusi.

Tes latensi jaringan

Tes ini mengirimkan 10 paket ke titik akhir yang sama dengan tes sebelumnya, dan memeriksa latensi paket. Semua paket harus memiliki latensi kurang dari 100 milidetik. Jika ada paket yang memiliki latensi lebih dari 100 milidetik, berkonsultasilah dengan insinyur jaringan untuk menentukan masalah dan menemukan solusinya.

Tes ukuran Unit Transmisi Maksimum (MTU)

Tes ini mendeteksi ukuran MTU dengan menggunakan alat Traceroute pada titik akhir yang sama dengan pengujian sebelumnya. Semua paket dalam tes harus memiliki ukuran MTU yang sama. Jika ada paket yang memiliki ukuran MTU yang berbeda, konsultasikan dengan spesialis sistem untuk menentukan masalah dan menemukan solusinya.

ID AMI menurut wilayah

Tabel berikut menunjukkan ID AMI untuk AMI Diagnostik AMS di wilayah Anda:

Wilayah	ID image
US East (Ohio)	ami-0f9c6801e8cab9399
US West (Oregon)	ami-01ceeab889d966e16
US East (N. Virginia)	ami-0617285c8257e8f6e
US West (N. California)	ami-0983b989d0a31ed55
Canada (Central)	ami-0ba9380b30eee288e
Europe (Ireland)	ami-05959fe4fcc448f01
Europe (London)	ami-0de91dfa18b365c04
Europe (Paris)	ami-0844ecd26f2cc9073
Europe (Frankfurt)	ami-09bc69e6da9f3adf4
Europe (Stockholm)	ami-01da340b09bf5ab78
Europe (Milan)	ami-0dcfc51c17cfe7727
Europe (Spain)	ami-0461c0b2e391c1280
Asia Pacific (Singapore)	ami-04150ffe00fe300d6
Asia Pacific (Sydney)	ami-049c1d1a9d72bafce
Asia Pacific (Jakarta)	ami-02189a4b8d3b2ee23
Asia Pacific (Hong Kong)	ami-0543ae06eff17cb07
Asia Pacific (Tokyo)	ami-07b91e8e25d7b7890
Asia Pacific (Seoul)	ami-0657ad8cb9b5cf71c
Asia Pacific (Osaka)	ami-01ccf3e974933afdc
Asia Pacific (Mumbai)	ami-0d9fa24bbb8225e60

Wilayah	ID image
Middle East (UAE)	ami-0e3dcf51dfe58929b
Middle East (Bahrain)	ami-0faad6f64101607f5
South America (São Paulo)	ami-097dd866777097d96

Tipe data AWS DMS	Deskripsi
STRING	Rangkaian karakter.
WSTRING	Rangkaian karakter double-byte.
BOOLEAN	Nilai Boolean.
BYTE	Nilai data biner.
TANGGAL	Nilai tanggal: tahun, bulan, hari.
WAKTU	Nilai waktu: jam, menit, detik.
WAKTUTANGGAL	Nilai stempel waktu: tahun, bulan, hari, jam, menit, detik, detik pecahan. Detik pecahan menskalakan hingga maksimum 9 digit. Berikut support format: YYYY:MM:DD HH:MM:SS.F(9). Untuk Amazon S3 Select dan Amazon S3 Glacier Select, format tipe data yang berbeda. Untuk informasi selengkapnya, lihat deskripsi tipe data <code>timestamp</code> primitif di Jenis Data yang Didukung dari Panduan Pengguna Amazon Simple Storage Service.
INT1	Satu-byte, bilangan bulat bertanda.
INT2	Dua-byte, bilangan bulat bertanda.
INT4	Empat-byte, bilangan bulat bertanda.
INT8	Delapan-byte, bilangan bulat bertanda.
NUMERIK	Nilai numerik yang tepat dengan menskalakan dan presisi tetap.
REAL4	Satu presisi nilai titik mengambang.
REAL8	Presisi ganda nilai titik mengambang.

Tipe data AWS DMS	Deskripsi
UINT1	Satu-byte, bilangan bulat tidak bertanda.
UINT2	Dua-byte, bilangan bulat tidak bertanda.
UINT4	Empat-byte, bilangan bulat tidak bertanda.
UINT8	Delapan-byte, bilangan bulat tidak bertanda.
BLOB	Objek large biner.
CLOB	Objek large karakter.
NCLOB	Objek large karakter asli.

Note

AWS DMS tidak dapat melakukan migrasi tipe data LOB ke titik akhir Apache Kafka.

AWS Catatan rilis DMS

Berikut ini, Anda dapat menemukan catatan rilis untuk versi AWS Database Migration Service (AWS DMS) saat ini dan sebelumnya.

AWS DMS tidak membedakan antara versi mayor dan minor saat Anda mengaktifkan pemutakhiran versi Otomatis untuk instance replikasi Anda. DMS secara otomatis memutakhirkan versi instance replikasi selama jendela pemeliharaan jika versi tersebut tidak digunakan lagi.

Perhatikan bahwa untuk memutakhirkan versi instance replikasi Anda secara manual (menggunakan API atau CLI) dari versi 3.4.x ke 3.5.x, Anda harus menyetel parameter ke `AllowMajorVersionUpgrade true` Untuk informasi tentang `AllowMajorVersionUpgrade` parameter, lihat [ModifyReplicationInstance](#) di dokumentasi DMS API.

Note

Versi mesin default saat ini untuk AWS DMS adalah 3.5.1.

Tabel berikut menunjukkan tanggal berikut untuk versi DMS aktif:

- Tanggal rilis versi
- Tanggal setelah itu Anda tidak dapat membuat instance baru dengan versi
- Tanggal ketika DMS secara otomatis memperbarui instance versi tersebut (tanggal EOL)

Versi	Tanggal rilis	Tidak ada tanggal instans baru	Tanggal EOL
3.5.2	Okt 29, 2023	30 Mar 2025	Apr 29, 2025
3.5.1	30 Jun 2023	Nov 30, 2024	Jan 30, 2025
3.4.7	31 Mei 2022	30 Jul 2024	Agustus 29, 2024
3.4.6	November 30, 2021	26 Mei 2024	Juni 27, 2024

AWS Catatan rilis Database Migration Service 3.5.2

Fitur baru di AWS DMS 3.5.2

Fitur baru atau perangkat tambahan	Deskripsi
Validasi data Redshift	AWS DMS sekarang mendukung validasi data dalam target Redshift.
Support untuk Microsoft SQL Server versi 2022 sebagai sumber dan target.	AWS DMS sekarang mendukung penggunaan Microsoft SQL Server versi 2022 sebagai sumber dan target.
IBM Db2 LUW sebagai target	AWS DMS sekarang mendukung IBM Db2 LUW sebagai target. Dengan menggunakan AWS DMS, Anda sekarang dapat melakukan migrasi langsung dari IBM Db2 LUW ke IBM Db2 LUW.

AWS Catatan rilis Database Migration Service 3.5.1

Tabel berikut menunjukkan fitur dan penyempurnaan baru yang diperkenalkan di AWS Database Migration Service (AWS DMS) versi 3.5.1.

Fitur baru atau perangkat tambahan	Deskripsi
Dukungan untuk PostgreSQL 15.x	AWS DMS versi 3.5.1 mendukung PostgreSQL versi 15.x. Lihat informasi yang lebih lengkap di Menggunakan PostgreSQL sebagai sumber dan Menggunakan PostgreSQL sebagai target .
Support untuk Amazon DocumentDB Elastic Clusters dengan koleksi sharded	AWS DMS versi 3.5.1 mendukung Amazon DocumentDB Elastic Clusters dengan koleksi sharded. Untuk informasi selengkapnya, lihat Menggunakan Amazon DocumentDB sebagai target untuk Database Migration Service AWS .
Redshift Tanpa Server sebagai Target	Support untuk menggunakan Amazon Redshift Serverless sebagai titik akhir target. Untuk informasi selengkapnya, lihat Menggunakan

Fitur baru atau perangkat tambahan	Deskripsi
	basis data Amazon Redshift sebagai target untuk AWS Database Migration Service.
Pengaturan Titik Akhir Babelfish	Pengaturan titik akhir target PostgreSQL yang disempurnakan untuk memberikan dukungan Babelfish. Untuk informasi selengkapnya, lihat Menggunakan basis data PostgreSQL sebagai target untuk AWS Database Migration Service.
Transaksi Terbuka Sumber Oracle	AWS DMS 3.5.1 meningkatkan metodologi penanganan transaksi terbuka saat memulai tugas khusus CDC dari Posisi Awal untuk sumber Oracle. Untuk informasi selengkapnya, lihat <code>OpenTransactionWindow</code> di bagian Pengaturan titik akhir saat menggunakan an Oracle sebagai sumber AWS DMS.
Amazon Timestream sebagai Target	Support untuk menggunakan Amazon Timestream sebagai titik akhir target. Untuk informasi selengkapnya, lihat Menggunakan Amazon Timestream sebagai target AWS Database Migration Service.

AWS DMS versi 3.5.1 mencakup masalah yang diselesaikan berikut:

Masalah terselesaikan	Deskripsi
Oracle sebagai sumber yang menumbuhkan sesi tidak aktif	Memperbaiki masalah untuk sumber Oracle di mana tugas khusus CDC terus berkembang sesi tidak aktif, menghasilkan pengecualian berikut: <code>ORA-00020: maximum number of processes exceeded on the source database</code>
Mereplikasi perubahan UPDATE ke DocumentDB	Memperbaiki masalah untuk DocumentDB sebagai target di mana pernyataan UPDATE tidak direplikasi dengan benar dalam beberapa skenario.

Masalah terselesaikan	Deskripsi
Tugas hanya validasi	Peningkatan penanganan kesalahan untuk fitur validasi data agar gagal tugas dengan benar saat validasi data dinonaktifkan untuk tugas validasi saja.
Replikasi pergeseran merah setelah penghentian koneksi	Memperbaiki masalah untuk target Redshift di mana tugas DMS tidak akan mencoba lagi menerapkan perubahan pada target ketika target telah <code>ParallelApplyThreads</code> menetapkan lebih besar dari nol setelah penghentian koneksi, yang akan mengakibatkan kehilangan data.
Teks MySQL ke replikasi <code>mediumtext</code>	Memperbaiki masalah untuk MySQL ke replikasi MySQL dari tipe data <code>mediumtext</code> dengan mode <code>Full-lob</code> .
Tugas CDC tidak bereplikasi dengan rahasia yang diputar	Memperbaiki masalah untuk tugas DMS dengan <code>BatchApplyEnabled</code> disetel ke <code>true</code> tempat DMS akan berhenti mereplikasi data setelah <code>Secrets Manager</code> memutar kata sandi.
Masalah segmentasi MongoDB/DocumentDB	Memperbaiki masalah untuk sumber MongoDB/DocDB di mana segmentasi rentang tidak akan berfungsi dengan baik ketika kolom kunci utama berisi nilai besar.
Validasi data Oracle dari nilai numerik yang tidak terikat	Memperbaiki masalah untuk target Oracle di mana DMS akan mengenali nilai tipe data yang tidak terikat <code>NUMERIC</code> sebagai <code>STRING</code> selama validasi data.
Validasi data SQL Server	Memperbaiki masalah untuk titik akhir SQL Server di mana validasi data DMS membangun pernyataan SQL yang tidak valid.
MongoDB Auto-Segmentasi	Meningkatkan fungsionalitas partisi otomatis data saat memigrasi dokumen secara paralel dari MongoDB sebagai sumber.
Format Parquet Amazon S3 Apache	Memperbaiki masalah sehingga file Apache Parquet yang ditulis ke S3 sebagai target dapat dilihat dengan Python dengan Apache Arrow C++.
PostgreSQL sebagai sumber penanganan DDL	Memperbaiki masalah dengan sumber PostgreSQL di mana operasi DDL yang tidak didukung tidak diabaikan dengan benar.

Masalah terselesaikan	Deskripsi
Kesalahan data PostgreSQL timestampz	Memperbaiki masalah migrasi PostgreSQL ke PostgreSQL di mana stempel waktu dengan data zona waktu tidak dimigrasi dengan benar dengan Batch apply diaktifkan selama CDC.
Kegagalan validasi Oracle ke PostgreSQL	Memperbaiki masalah migrasi Oracle ke PostgreSQL di mana validasi data gagal untuk tipe data NUMERIC (38,30).
Kesalahan tipe data yang diperluas Oracle	Memperbaiki masalah dengan sumber Oracle di mana tipe data varchar yang diperluas sedang dipotong.
Menggabungkan operator filter	Memperbaiki masalah untuk fungsionalitas pemfilteran kolom di mana operator kolom null tidak dapat digabungkan dengan jenis operator lainnya.
Latensi CDC yang dihasilkan dari logging yang berlebihan.	Memperbaiki masalah dengan sumber PostgreSQL di mana logging peringatan plug-in pglogical yang berlebihan menyebabkan latensi CDC sumber.
Penanganan replikasi dua arah dari Create Table DDL	Memperbaiki masalah replikasi dua arah dari PostgreSQL ke PostgreSQL di mana perubahan Create Table DDL tidak direplikasi dengan benar.
Kegagalan CDC saat menggunakan filter	Memperbaiki masalah untuk fitur pemfilteran di mana replikasi CDC gagal.
Validasi nama host otoritas sertifikat untuk titik akhir Kafka	Meningkatkan fungsionalitas titik akhir Kafka dengan menambahkan opsi untuk menonaktifkan validasi nama host dari otoritas sertifikat (<code>().SslEndpointIdentificationAlgorithm</code>
Validasi IBM Db2 LUW	Memperbaiki masalah saat tanggal sumber, stempel waktu, dan tipe data waktu Db2 LUW tidak ditangani dengan benar selama validasi data.
Validasi S3	Memperbaiki masalah dengan migrasi Db2 LUW ke S3 di mana fungsi validasi tidak menangani tipe data stempel waktu (0) dengan benar.

Masalah terselesaikan	Deskripsi
Kegagalan restart tugas DMS	Memperbaiki masalah dengan sumber PostgreSQL di mana AWS DMS tugas gagal memulai ulang dan tidak dapat menggunakan peristiwa relasional saat menggunakan plugin pglogical.
SQL Server validasi tipe data HIERARKI	Memperbaiki masalah untuk sumber SQL Server di mana validasi tipe data HIERARKI akan gagal.
String SQL Server dengan karakter kontrol	Memperbaiki masalah untuk SQL Server di mana string dengan karakter kontrol tidak direplikasi dengan benar.
Redshift dengan Secrets Manager	Memperbaiki masalah dengan target Redshift di mana pengujian titik akhir akan gagal saat menggunakan Secrets Manager.
Inkonsistensi pengaturan ParallelLoadThreads MySQL	Memperbaiki masalah dengan target MySQL di mana ParallelLoadThreads pengaturan tidak dipertahankan dengan benar setelah pengaturan tugas berubah.
Kesalahan dengan PostgreSQL ke pemetaan tipe data Oracle	Memperbaiki masalah dengan PostgreSQL ke migrasi Oracle di mana tugas akan gagal saat mereplikasi dari tipe data TEXT ke tipe data VARCHAR2 (2000).
Validasi data Oracle ke PostgreSQL	Memperbaiki masalah dengan migrasi Oracle ke PostgreSQL di mana validasi data melaporkan positif palsu ketika karakter NULL direplikasi sebagai karakter SPACE.
Sumber SQL Server dalam konfigurasi AlwaysOn	Memperbaiki masalah dengan sumber SQL Server dalam AlwaysOn kebingungan di mana AWS DMS tugas akan gagal ketika nama replika tidak sama persis dengan nama server yang sebenarnya.
Kegagalan pengujian titik akhir sumber Oracle	Memperbaiki masalah dengan sumber Oracle di mana tes koneksi AWS DMS titik akhir akan gagal karena privileges yang tidak mencukupi saat mengambil ID sesi Oracle (SID).
CDC tidak mengambil tabel baru	Memperbaiki masalah dengan tugas khusus CDC di mana tabel yang dibuat pada sumber setelah tugas dimulai tidak direplikasi dalam beberapa kasus.

Masalah terselesaikan	Deskripsi
Buka transaksi di Oracle sebagai sumber	Meningkatkan metodologi penanganan transaksi terbuka saat memulai tugas khusus CDC dari Posisi Awal untuk sumber Oracle.
Masalah data yang hilang	Memperbaiki masalah data yang hilang saat melanjutkan tugas jika dihentikan setelah perubahan cache diterapkan (<code>StopTaskCachedChangesApplied</code> opsi disetel ke <code>true</code>). Masalah ini jarang terjadi jika AWS DMS tetap ada perubahan cache pada disk instance AWS DMS replikasi karena volume perubahan yang tinggi pada sumbernya.
Masalah validasi data pada tipe data yang diperluas	Memperbaiki masalah untuk PostgreSQL ke validasi data Oracle di mana validasi gagal untuk tipe data yang diperluas.
Masalah validasi data pada pengkodean karakter yang tidak konsisten	Memperbaiki masalah untuk SQL Server untuk validasi data PostgreSQL di mana validasi gagal ketika pengkodean karakter tidak konsisten antara sumber dan target.
Masalah validasi data ORA-01455	Memperbaiki masalah saat terjadi kesalahan ORA-01455 selama validasi saat <code>integer</code> PostgreSQL memetakan ke Oracle. <code>number(10)</code>
Dukungan SQL Server IDENTITY	Memperbaiki masalah replikasi data SQL Server ke SQL Server di mana kolom identitas migrasi gagal saat kolom target memiliki properti. <code>IDENTITY</code>
Masalah set karakter dengan pernyataan ALTER	Memperbaiki masalah AWS DMS replikasi MySQL ke MySQL di mana mengubah karakter yang disetel ke UTF16 saat memigrasi kan pernyataan selama CDC. <code>ALTER</code>
Dukungan tipe data PostgreSQL ke Redshift Spasial	Menambahkan dukungan untuk <code>spatial</code> tipe data saat bermigrasi dari PostgreSQL ke Amazon Redshift.
Kompresi GZIP dari file.parquet	Memperbaiki masalah saat AWS DMS gagal menghasilkan file.parquet dengan kompresi GZIP dengan S3 sebagai target.

Masalah terselesaikan	Deskripsi
Migrasi sumber MongoDB/DocDB	Memperbaiki masalah saat AWS DMS tidak memigrasikan beberapa partisi dari sumber MongoDB.
Masalah statistik tabel	Memperbaiki masalah ketika statistik tabel tidak ditampilkan ketika setidaknya salah satu tugas pada instance replikasi berisi lebih dari 1001 tabel.
Tabel ditangguhkan untuk IBM Db2 LUW versi 10.1.0 dan lebih rendah	Memperbaiki masalah untuk sumber Db2 LUW di mana migrasi tabel ditangguhkan dengan kesalahan TYPESTRINGUNITS is not valid saat versi basis data sumber 10.1.0 atau lebih rendah.
Masalah partisi MongoDB	Memperbaiki masalah untuk MongoDB/DocDB di mana satu atau lebih segmen partisi sumber hilang.
Masalah partisi MongoDB	Perbaiki masalah saat segmentasi berdasarkan kolom dengan tipe NumberLong () gagal karena bug konversi tipe.
Masalah partisi MongoDB	Peningkatan kinerja autosegmentasi untuk kumpulan data besar dengan MongoDB sebagai sumber.
Versi driver MongoDB	Menurunkan versi driver MongoDB ke 1.20.0 untuk melanjutkan dukungan untuk MongoDB versi 3.6 dan yang lebih rendah.
Jenis data stempel waktu Parquet Amazon S3 Apache	Memperbaiki masalah untuk target parquet Amazon S3. AWS DMS sekarang menetapkan parameter format isAdjustedToUTC true agar sesuai dengan perilaku di versi sebelumnya AWS DMS.
Amazon Redshift sebagai perintah salin target	Memperbaiki masalah Amazon Redshift sebagai target di mana perintah salin gagal untuk tabel besar saat menyalin data dari Amazon S3 ke Amazon Redshift.
Tipe data geometri PostgreSQL	Memperbaiki masalah migrasi PostgreSQL ke PostgreSQL di mana migrasi gagal pada tipe data geometri besar.
Oracle ke PostgreSQL XML	Memperbaiki masalah saat migrasi menambahkan ruang ekstra pada XHTML saat mereplikasi dari Oracle ke PostgreSQL.

Masalah terselesaikan	Deskripsi
Memperbarui pos pemeriksaan target di mesin yang didukung	AWS DMS sekarang memperbarui pos pemeriksaan target dalam <code>awsdms_txn_state</code> tabel di database target.
Catatan MongoDB/DocDB dikirim ke koleksi yang salah	Memperbaiki masalah untuk MongoDB/DocDB di mana data dikirim ke koleksi target yang salah.
Pilihan tabel baru sumber Oracle dengan pengaturan titik EscapeCharacter akhir	Memperbaiki masalah untuk sumber Oracle di mana AWS DMS hanya mengambil tabel baru untuk replikasi ketika tugas dihentikan dan dilanjutkan saat pengaturan titik EscapeCharacter akhir ditetapkan.
Pos pemeriksaan pemulihan CDC	Memperbaiki inkonsistensi di pos pemeriksaan pemulihan CDC yang diamati antara datastore target dan konsol. AWS DMS
Tugas hanya validasi CDC	Memperbaiki masalah dengan tugas khusus validasi CDC di mana tugas tidak akan gagal, bahkan jika semua tabel dalam tugas mengalami kegagalan.
Perilaku validasi dengan masalah koneksi sumber atau target	Memperbaiki masalah dengan validasi data di mana AWS DMS akan menanggukkan tabel pada sumber atau target saat koneksi terputus.
Oracle ke PostgreSQL validasi data positif palsu	Memperbaiki masalah dengan validasi data Oracle ke PostgreSQL jika dilaporkan positif palsu. AWS DMS ini karena perbedaan representasi karakter NULL sumber pada target tidak diperhitungkan dengan tipe data berbasis teks selain VARCHAR.
Oracle ke PostgreSQL pemotongan data	Memperbaiki masalah dengan Oracle sebagai sumber dan PostgreSQL sebagai target AWS DMS di mana memotong data untuk kolom NVARCHAR dengan pengaturan Oracle disetel ke <code>NLS_NCHAR_CHARACTERSET AL16UTF16</code>

Masalah terselesaikan	Deskripsi
Kesalahan validasi data	Memperbaiki masalah dengan validasi data di mana <code>unable to create where filter clause</code> kesalahan dilemparkan saat pemfilteran sumber dan aturan transformasi kolom tambahan sedang digunakan.
Penanganan kesalahan target Redshift	Memperbaiki masalah dengan Redshift sebagai target di mana penanganan kesalahan tidak berfungsi seperti yang dikonfigurasi saat tugas CDC mengatur pengaturan <code>ParallelApplyThreads</code> tugas ke nilai yang lebih besar dari nol.
Oracle sebagai kegagalan komunikasi sumber	Memperbaiki masalah dengan Oracle sebagai sumber di mana tugas tetap dalam <code>RUNNING</code> status, tetapi tidak dapat memigrasikan data apa pun setelah kegagalan komunikasi.
Tabel CDC ditangguhkan dengan filter kolom	Memperbaiki masalah dengan tugas Full Load+CDC di mana tabel akan ditangguhkan selama fase CDC saat filter kolom diterapkan.
S3 sebagai kegagalan validasi data target untuk karakter khusus	Memperbaiki masalah dengan validasi data target S3 di mana tugas akan gagal jika nama tabel menyertakan karakter khusus selain garis bawah.
Sumber MongoDB Beban Penuh dan kegagalan CDC	Memperbaiki masalah dengan MongoDB sebagai sumber di mana tugas Full Load+CDC akan gagal selama penanganan peristiwa cache saat memigrasi koleksi besar.
Tingkatkan masalah dengan <code>BatchApplyEnabled</code> disetel ke true	Memperbaiki masalah saat tugas dengan pengaturan <code>BatchApplyEnabled</code> tugas yang disetel ke true akan gagal setelah bermigrasi dari AWS DMS versi 3.4.6 ke 3.5.1 dalam beberapa kasus.
AlwaysOn Sumber SQL Server dengan pemeriksaan peka huruf besar/kecil	Memperbaiki masalah dengan SQL Server AlwaysOn sebagai sumber di mana tugas akan gagal dengan pemeriksaan peka huruf besar/kecil.

Masalah terselesaikan	Deskripsi
Tugas sumber MySQL menggantung	Memperbaiki masalah dengan MySQL sebagai sumber di mana tugas akan hang alih-alih gagal ketika sumber tidak dikonfigurasi dengan benar.
Kegagalan tugas beban penuh sumber S3	Memperbaiki masalah dengan S3 sebagai sumber di mana tugas akan gagal saat melanjutkan setelah memutakhirkan dari AWS DMS versi 3.4.6 atau 3.4.7 ke versi 3.5.1.
Sumber PostgreSQL dengan CaptureDdls disetel ke false	Memperbaiki masalah dengan PostgreSQL sebagai sumber di mana DDL tidak ditangani dengan benar dengan pengaturan titik akhir yang disetel ke false. CaptureDDLs
Tugas sumber Oracle mogok saat melanjutkan	Memperbaiki masalah dengan Oracle sebagai sumber di mana tugas akan macet saat resume karena data yang salah dalam nama kolom.
Kegagalan pencarian LOB sumber MySQL	Memperbaiki masalah dengan MySQL sebagai sumber di mana pencarian LOB akan gagal saat pengaturan tugas disetel ke nilai <code>ParallelApplyThreads</code> yang lebih besar dari nol.
Sumber SQL Server kesalahan LSN tidak logis	Memperbaiki masalah dengan SQL Server sebagai sumber di mana tugas akan gagal dengan <code>illogical LSN sequencing state error</code> kesalahan setelah memutakhirkan dari AWS DMS versi 3.4.7 ke versi 3.5.1.
Sumber PostgreSQL dengan pglogical	Memperbaiki masalah dengan PostgreSQL sebagai sumber di mana tugas menggunakan plugin pglogical akan gagal saat tugas dihentikan, tabel dihapus dari aturan pemilihan, tugas dilanjutkan, dan perubahan dilakukan pada tabel yang dihapus.
Aurora MySQL salah pos pemeriksaan pemulihan.	Memperbaiki masalah untuk Aurora MySQL sebagai sumber di mana pos pemeriksaan pemulihan yang salah akan disimpan sebagai akibat dari failover Aurora atau sumber Aurora berhenti dan mulai.

Masalah terselesaikan	Deskripsi
SQL Server sebagai crash tugas sumber.	Memperbaiki masalah untuk SQL Server sebagai sumber di mana tugas akan macet saat <code>SafeguardPolicy</code> disetel ke <code>RELY_ON_SQL_SERVER_REPLICATION_AGENT</code> .
Pengecoran tipe data salah dengan MySQL sebagai target	Memperbaiki masalah untuk MySQL sebagai target di sana di mana replikasi CDC akan gagal sebagai akibat dari pengecoran tipe data yang salah dalam fase penerapan batch.
Kegagalan tugas dengan <code>CaptureDdls</code> disetel ke <code>false</code> untuk PostgreSQL sebagai sumber.	Memperbaiki masalah untuk PostgreSQL sebagai sumber di mana tugas akan gagal karena DDL diperlakukan sebagai DHTML ketika pengaturan titik akhir disetel ke <code>CaptureDDLs false</code> .
MongoDB crash koleksi kosong	Memperbaiki masalah untuk MongoDB sebagai sumber di mana tugas akan macet karena koleksi kosong.
Redshift sebagai target tugas beban penuh crash	Memperbaiki masalah untuk Redshift sebagai target di mana tugas akan macet selama fase pemuatan penuh saat tabel kontrol pos pemeriksaan pemulihan diaktifkan.
S3 ke S3 - tidak ada pergerakan data.	Memperbaiki masalah replikasi S3 ke S3 yang tidak AWS DMS akan mereplikasi data jika tidak ditentukan. <code>bucketFolder</code>
Latensi CDC dengan <code>GlueCatalogGeneration</code> disetel ke <code>true</code>	Memperbaiki masalah untuk S3 sebagai target di mana latensi berlebihan akan terjadi saat <code>GlueCatalogGeneration</code> disetel ke <code>true</code> .
Oracle sebagai pemotong data target	Memperbaiki masalah dengan Oracle sebagai target di mana AWS DMS memotong data di kolom <code>VARCHAR2</code> .
PostgreSQL menggaris bawahi perilaku wildcard	Memperbaiki masalah untuk PostgreSQL sebagai sumber di mana perilaku wildcard <code>'_'</code> dalam aturan pemilihan tidak berfungsi seperti yang didokumentasikan.
PostgreSQL sebagai sumber masalah header WAL kosong.	Memperbaiki masalah untuk PostgreSQL sebagai sumber di mana tugas akan gagal karena header WAL kosong yang diterima dari slot replikasi.

Masalah terselesaikan	Deskripsi
MySQL atau MariaDB sebagai sumber dengan log biner terkompresi	Memperbaiki masalah untuk MySQL dan MariaDB sebagai sumber di mana pesan kesalahan yang tepat tidak dipancarkan saat mendeteksi kompresi BINLOG. AWS DMS
Karakter khusus validasi data S3	Peningkatan validasi data S3 untuk menangani karakter khusus di kolom kunci primer dan non-primer.
Entri log tugas yang menyesatkan dengan Redshift sebagai target.	Memperbaiki masalah untuk Redshift sebagai target di mana keseluruhan yang menyesatkan hadir dalam log tugas yang melaporkan kegagalan pernyataan penerapan batch pada PEMBARUAN dan PENGHAPUSAN.
SQL Server ke tugas migrasi S3 crash.	Memperbaiki masalah migrasi SQL Server ke S3 di mana tugas akan macet saat menerapkan perubahan cache.
Data yang hilang pada kesalahan penerapan batch.	Memperbaiki masalah untuk fitur batch-apply di mana kesalahan dalam menerapkan batch akan mengakibatkan data hilang.

AWS Database Migration Service 3.5.0 Catatan rilis beta

Important

AWS DMS 3.5.0 adalah versi beta dari mesin instance replikasi. AWS DMS mendukung versi ini sama seperti semua rilis sebelumnya. Tetapi kami menyarankan Anda menguji AWS DMS 3.5.0 Beta sebelum menggunakannya untuk tujuan produksi.

Tabel berikut menunjukkan fitur dan penyempurnaan baru yang diperkenalkan di AWS Database Migration Service (AWS DMS) versi 3.5.0 Beta.

Fitur baru atau perangkat tambahan	Deskripsi
Perjalanan Waktu untuk Oracle dan Microsoft SQL Server	Anda sekarang dapat menggunakan Time Travel di semua AWS Wilayah dengan Oracle yang didukung DMS, Microsoft SQL Server, dan titik akhir sumber PostgreSQL, dan titik akhir target PostgreSQL dan MySQL yang didukung DMS.
Validasi S3	AWS DMS sekarang mendukung memvalidasi data yang direplikasi di titik akhir target Amazon S3. Untuk informasi tentang memvalidasi data target Amazon S3, lihat Validasi data target Amazon S3
Integrasi Katalog Glue	AWS Glue adalah layanan yang menyediakan cara sederhana untuk mengkategorikan data, dan terdiri dari repositori metadata yang dikenal sebagai AWS Glue Data Catalog. Anda sekarang dapat mengintegrasikan AWS Glue Data Catalog dengan titik akhir target Amazon S3 Anda dan kueri data Amazon S3 melalui layanan lain AWS seperti Amazon Athena. Untuk informasi selengkapnya, lihat Menggunakan AWS Glue Data Catalog dengan target Amazon S3 untuk AWS DMS .
Paralel berlaku untuk DocumentDB sebagai target	Menggunakan DocumentDB sebagai target dengan pengaturan tugas ParallelApply* baru AWS DMS, sekarang mendukung maksimum 5000 catatan per detik selama replikasi CDC. Untuk informasi selengkapnya, lihat Menggunakan Amazon DocumentDB sebagai target untuk Database Migration Service AWS .
Customer Centric Logging	Anda sekarang dapat memeriksa dan mengelola log tugas secara lebih efektif dengan AWS DMS versi 3.5.0. Untuk informasi tentang melihat dan mengelola log tugas AWS DMS, lihat Melihat dan mengelola AWS Mencatat tugas DMS .
Mekanisme SASL_PLAIN untuk titik akhir target Kafka	Anda sekarang dapat menggunakan otentikasi SASL_PLAIN untuk mendukung titik akhir target Kafka MSK.
Replikasi transaksi XA di MySQL	Anda sekarang dapat menggunakan transaksi XA pada sumber MySQL DMS Anda. Sebelum DMS 3.5.0, perubahan DHTML

Fitur baru atau perangkat tambahan	Deskripsi
	diterapkan sebagai bagian dari transaksi XA tidak direplikasi dengan benar.
Jenis Data Diperpanjang Oracle	AWS DMS sekarang mendukung replikasi tipe data yang diperluas di Oracle versi 12.2 dan yang lebih tinggi.
Db2 LUW Lingkungan PureScale	AWS DMS sekarang mendukung replikasi dari lingkungan Db2 LUW. PureScale Fungsionalitas ini hanya didukung menggunakan opsi Mulai memproses perubahan dari posisi perubahan sumber.
Sumber SQL Server dengan opsi READ_COMMITTED_SNAPSHOT	Saat menggunakan database sumber Microsoft SQL Server dengan READ_COMMITTED_SNAPSHOT opsi yang disetel keTRUE, Anda dapat mereplikasi perubahan DHTML dengan benar dengan mengatur atribut koneksi forceDataRowPencarian.

AWS DMS 3.5.0 mencakup masalah yang diselesaikan berikut:

Masalah teratasi di AWS DMS 3.5.0 diluncurkan pada 17-Maret-2023

Topik	Penyelesaian
Oracle — membandingkan kasus khusus untuk string yang dikonversi dari numerik	Memperbaiki masalah untuk sumber Oracle di mana aturan pemfilteran tidak berfungsi seperti yang diharapkan untuk kolom numerik ketika transformasi tipe data ke string ada untuk kolom yang sama.
Penyempurnaan SQL Server AG lokal	Peningkatan efisiensi penanganan koneksi dengan sumber SQL Server dalam AlwaysOn konfigurasi dengan menghilangkan koneksi yang tidak perlu ke replika yang tidak digunakan oleh DMS.
Konversi internal SQL Server HIERARCHY ID	Memperbaiki masalah dengan SQL Server Source di mana tipe data HIERARCHYID direplikasi sebagai ID

Topik	Penyelesaian
	VARCHAR (250) alih-alih HIERARCHYID ke target SQL Server.
Perbaikan tugas pemindahan target S3	Memperbaiki masalah saat memindahkan tugas dengan target S3 akan memakan waktu yang sangat lama, tampak beku atau tidak pernah selesai.
Mekanisme Dataran SASL Kafka	Memperkenalkan dukungan untuk metode otentikasi SASL Plain untuk titik akhir target Kafka MSK.
Beban/Terapkan Paralel gagal karena parameter <code>_type</code> dengan Opensearch 2.x	Memperbaiki masalah untuk target Opensearch 2.x di mana beban paralel atau penerapan paralel akan gagal karena kurangnya dukungan untuk parameter <code>_type</code> .
Support filter pemetaan tabel dengan operator campuran	Menghapus batasan di mana hanya satu filter yang dapat diterapkan pada kolom.
S3, Kinesis, titik akhir Kafka — migrasi kolom lob berbasis perubahan dalam fase CDC	Memperbaiki masalah untuk target Kinesis, Kafka, dan S3 di mana data di kolom LOB yang ditambahkan selama CDC tidak direplikasi.
Peningkatan driver MongoDB	Upgrade driver MongoDB ke v1.23.2.
Pembaruan driver Kafka	Upgrade driver Kafka dari 1.5.3 ke 1.9.2.
Pengaturan titik akhir S3 tidak berfungsi dengan baik	Memperbaiki masalah untuk target S3 di mana pengaturan <code>AddTrailingPaddingCharacter</code> titik akhir tidak berfungsi saat data berisi karakter yang ditentukan sebagai pembatas untuk target S3.

Topik	Penyelesaian
Tugas target Kinesis akan macet	Memperbaiki masalah untuk target Kinesis di mana tugas akan macet saat nilai PK kosong dan debug terperinci diaktifkan.
Ketika nama kolom target S3 digeser oleh satu posisi	Memperbaiki masalah untuk target S3 di mana nama kolom digeser oleh satu posisi saat AddCoLumnName disetel true dan TimestampCoLumnName disetel ke. ""
Peringatan pemotongan LOB logging yang ditingkatkan	Peningkatan logging peringatan untuk pemotongan LOB untuk sumber SQL Server untuk menyertakan pernyataan pilih yang digunakan untuk mengambil LOB.
Tambahkan kesalahan Fatal untuk menghindari kerusakan tugas DMS jika kata sandi TDE salah.	Memperkenalkan pesan kesalahan yang berarti dan menghilangkan masalah kerusakan tugas dalam situasi di mana tugas DMS gagal tanpa pesan kesalahan karena kata sandi TDE yang salah untuk Oracle sebagai sumber.
Mengizinkan PostgreSQL CTAS (Buat tabel seperti yang dipilih) migrasi DDL selama CDC.	Keterbatasan DMS yang dihapus tidak dapat mereplikasi PostgreSQL CTAS (membuat tabel sebagai dipilih) DDL selama CDC.
Perbaiki crash tugas pg_logical saat kolom tabel dijatuhkan di CDC.	Memperbaiki masalah untuk sumber PostgreSQL dengan target S3 di mana kolom tidak sejajar pada target saat dukungan untuk LOB dinonaktifkan dan LOB hadir.
Perbaiki kebocoran memori dalam penanganan koneksi MySQL	Memperbaiki masalah untuk sumber MySQL di mana konsumsi memori tugas terus meningkat.

Topik	Penyelesaian
Pengaturan titik akhir sumber Oracle - <code>ConvertTimestampWithZoneToUTC</code>	Tetapkan atribut ini <code>true</code> untuk mengonversi nilai stempel waktu kolom 'TIMESTAMP WITH TIME ZONE' dan 'TIMESTAMP WITH LOCAL TIME ZONE' ke UTC. Secara default nilai atribut ini adalah 'false' dan data direplikasi menggunakan zona waktu database sumber.
Sumber Oracle - <code>DataTruncationErrorPolicy</code> ke <code>SUSPEND_TABLE</code> tidak berfungsi	Memperbaiki masalah untuk sumber Oracle dengan target S3 di mana tabel tidak ditangguhkan saat pengaturan <code>DataTruncationErrorPolicy</code> tugas disetel ke <code>SUSPEND_TABLE</code> .
SQL Server gagal pada skema/tabel panjang saat membangun klausa kueri	Memperbaiki masalah untuk sumber SQL Server di mana tugas akan gagal atau menjadi tidak responsif ketika aturan pemilihan berisi daftar tabel yang dipisahkan koma.
Otentikasi Secret Manager dengan titik akhir MongoDB	Memperbaiki masalah untuk titik akhir MongoDB dan DocumentDB di mana otentikasi berbasis manajer rahasia tidak berfungsi.
DMS memotong data selama CDC untuk kolom <code>varchar</code> multi-byte saat <code>NLS_NCHAR_CHARACTERSET</code> diatur ke UTF8	Memperbaiki masalah untuk sumber Oracle dengan target Oracle di mana data sedang dipotong untuk kolom <code>VARCHAR</code> multi-byte dengan <code>NLS_NCHAR_CHARACTERSET</code> diset ke UTF8.
<code>filterTransactionsOfUser</code> ECA untuk Oracle LogMiner	Menambahkan Extra Connection Attribute (ECA) <code>filterTransactionsOfUser</code> untuk memungkinkan DMS mengabaikan transaksi dari pengguna tertentu saat mereplikasi dari Oracle menggunakan LogMiner

Topik	Penyelesaian
SQL Server Mengatur kesalahan yang dapat dipulihkan saat lsn hilang dari cadangan	Memperbaiki masalah untuk SQL Server di mana tugas tidak akan gagal pada LSN yang hilang.

AWS Catatan rilis Database Migration Service 3.4.7

Tabel berikut menunjukkan fitur dan penyempurnaan baru yang diperkenalkan di AWS Database Migration Service (AWS DMS) versi 3.4.7.

Fitur baru atau perangkat tambahan	Deskripsi
Support Babelfish sebagai target	<p>AWS DMS sekarang mendukung Babelfish sebagai target. Dengan menggunakan AWS DMS, Anda sekarang dapat memigrasikan data langsung dari sumber apa pun yang AWS DMS didukung ke Babelfish, dengan waktu henti minimal.</p> <p>Untuk informasi selengkapnya, lihat Menggunakan Babelfish sebagai target untuk AWS Database Migration Service.</p>
Support IBM Db2 z/OS database sebagai sumber untuk full load saja	<p>AWS DMS sekarang mendukung database IBM Db2 z/OS sebagai sumber. Dengan menggunakan AWS DMS, Anda sekarang dapat melakukan migrasi langsung dari mainframe Db2 ke target yang didukung. AWS DMS</p> <p>Untuk informasi selengkapnya, lihat Menggunakan IBM Db2 untuk database z/OS sebagai sumber untuk AWS DMS.</p>
Support SQL Server membaca replika sebagai sumber	<p>AWS DMS sekarang mendukung SQL Server membaca replika sebagai sumber. Dengan menggunakan AWS DMS, Anda sekarang dapat melakukan migrasi langsung dari replika baca SQL Server ke target yang AWS DMS didukung.</p>

Fitur baru atau perangkat tambahan	Deskripsi
	<p>Untuk informasi selengkapnya, lihat Menggunakan database Microsoft SQL Server sebagai sumber AWS DMS.</p>
Mendukung EventBridge acara DMS	<p>AWS DMS mendukung pengelolaan langganan acara yang digunakan EventBridge untuk acara DMS.</p> <p>Untuk informasi selengkapnya, lihat Bekerja dengan Amazon EventBridge tindakan dan notifikasi diAWS Database Migration Service.</p>
Support sumber VPC dan titik akhir target	<p>AWS DMS sekarang mendukung titik akhir Amazon Virtual Private Cloud (VPC) sebagai sumber dan target. AWS DMS sekarang dapat terhubung ke AWS layanan apa pun dengan titik akhir VPC ketika rute yang ditentukan secara eksplisit ke layanan ditentukan dalam VPC mereka. AWS DMS</p> <div data-bbox="544 955 1507 1417" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"><p> Note</p><p>Upgrade ke AWS DMS versi 3.4.7 dan yang lebih tinggi mengharuskan Anda mengonfigurasi terlebih dahulu AWS DMS untuk menggunakan titik akhir VPC atau menggunakan rute publik. Persyaratan ini berlaku untuk titik akhir sumber dan target untuk Amazon S3, Amazon Kinesis Data Streams, Amazon AWS Secrets Manager DynamoDB, Amazon Redshift, dan Amazon Service. OpenSearch</p></div> <p>Untuk informasi selengkapnya, lihat Mengkonfigurasi titik akhir VPC sebagaiAWSSumber DMS dan titik akhir target.</p>
Versi PostgreSQL baru	<p>PostgreSQL versi 14.x sekarang didukung sebagai sumber dan sebagai target.</p>

Fitur baru atau perangkat tambahan	Deskripsi
Support Aurora Serverless v2 sebagai target	<p>AWS DMS sekarang mendukung Aurora Serverless v2 sebagai target. Dengan menggunakan AWS DMS, Anda sekarang dapat melakukan migrasi langsung ke Aurora Tanpa Server v2.</p> <p>Untuk informasi tentang AWS DMS target yang didukung, lihat Target migrasi data.</p>
IBM Db2 baru untuk versi LUW	<p>AWS DMS sekarang mendukung IBM Db2 untuk LUW versi 11.5.6 dan 11.5.7 sebagai sumber. Dengan menggunakan AWS DMS, Anda sekarang dapat melakukan migrasi langsung dari versi terbaru IBM DB2 untuk LUW.</p> <p>Untuk informasi tentang AWS DMS sumber, lihat Sumber untuk migrasi data.</p> <p>Untuk informasi tentang AWS DMS target yang didukung, lihat Target migrasi data.</p>

AWS DMS 3.4.7 mencakup perilaku baru atau perubahan berikut dan masalah yang diselesaikan:

- Anda sekarang dapat menggunakan format tanggal dari definisi tabel untuk mengurai string data menjadi objek tanggal saat menggunakan Amazon S3 sebagai sumber.
- Penghitung statistik tabel baru sekarang tersedia: `AppliedInserts`, `AppliedDdls`, `AppliedDeletes`, dan `AppliedUpdates`.
- Anda sekarang dapat memilih jenis pemetaan default saat menggunakan OpenSearch sebagai target.
- Pengaturan `TrimSpaceInChar` endpoint baru untuk sumber Oracle, PostgreSQL, dan SQLServer memungkinkan Anda menentukan apakah akan memangkas data pada tipe data CHAR dan NCHAR.
- Pengaturan `ExpectedBucketOwner` endpoint baru untuk Amazon S3 mencegah sniping saat menggunakan S3 sebagai sumber atau target.
- Untuk RDS SQL Server, Azure SQL Server, dan SQL Server yang dikelola sendiri - DMS sekarang menyediakan penyiapan otomatis MS-CDC pada semua tabel yang dipilih untuk tugas

migrasi yang dengan atau tanpa KUNCI PRIMARY, atau dengan indeks unik mengingat prioritas pengaktifan untuk MS-REPLICATION pada tabel SQL Server yang dikelola sendiri dengan KUNCI PRIMER.

- Menambahkan dukungan untuk replikasi Oracle Partition dan sub-partisi Operasi DDL selama migrasi homogen Oracle.
- Memperbaiki masalah saat tugas validasi data macet dengan kunci primer komposit saat menggunakan Oracle sebagai sumber dan target.
- Memperbaiki masalah dengan mentransmisikan tipe karakter yang bervariasi dengan benar ke boolean saat kolom target dibuat sebelumnya sebagai boolean saat menggunakan Redshift sebagai target.
- Memperbaiki masalah yang menyebabkan pemotongan data untuk tipe `varchar` data yang dimigrasikan `varchar(255)` karena masalah ODBC yang diketahui saat menggunakan PostgreSQL sebagai target.
- Memperbaiki masalah di mana Petunjuk Paralel untuk operasi DELETE tidak dihormati dengan `BatchApplyEnabled` disetel ke `true` dan `BatchApplyPreserveTransaction` disetel ke `false` saat menggunakan Oracle sebagai target.
- Pengaturan `AddTrailingPaddingCharacter` endpoint baru untuk Amazon S3 menambahkan padding pada data string saat menggunakan S3 sebagai target.
- Pengaturan `max_statement_timeout_seconds` tugas baru memperpanjang batas waktu default dari kueri titik akhir. Pengaturan ini saat ini digunakan oleh kueri metadata titik akhir MySQL.
- Saat menggunakan PostgreSQL sebagai target, memperbaiki masalah di mana tugas CDC tidak memanfaatkan pengaturan tugas penanganan kesalahan dengan benar.
- Memperbaiki masalah saat DMS tidak dapat mengidentifikasi mode Redis dengan benar untuk instance Redis Enterprise.
- Memperluas dukungan atribut koneksi `includeOpForFullLoad` tambahan (ECA) untuk format `parquet` target S3.
- Memperkenalkan pengaturan titik akhir PostgreSQL baru. `migrateBooleanAsBoolean` Saat pengaturan ini disetel ke `true` migrasi PostgreSQL ke Redshift, boolean akan dimigrasikan sebagai `varchar(1)`. Ketika disetel ke `false`, boolean dimigrasikan sebagai `varchar(15)`, yang merupakan perilaku default.
- Saat menggunakan sumber SQL Server, memperbaiki masalah migrasi dengan `datetime` tipe data. Perbaikan ini mengatasi masalah penyisipan `Null` saat presisi dalam milidetik.

- Untuk sumber PostgreSQL dengan PGLOGICAL, memperbaiki masalah migrasi saat menggunakan pglogical dan menghapus bidang dari tabel sumber selama fase CDC, di mana nilai setelah bidang yang dihapus tidak dimigrasikan ke tabel target.
- Memperbaiki masalah migrasi SQL Server Loopback dengan replikasi dua arah mendapatkan catatan berulang.
- Menambahkan ECA baru `mapBooleanAsBoolean` untuk PostgreSQL sebagai sumber. Menggunakan atribut koneksi tambahan ini, Anda dapat mengganti pemetaan tipe data default dari PostgreSQL Boolean ke tipe data Boolean. RedShift
- Memperbaiki masalah migrasi saat menggunakan SQL Server sebagai sumber yang menangani ALTER DECIMAL/NUMERIK SCALE yang tidak mereplikasi ke target.
- Memperbaiki masalah koneksi dengan SQL Server 2005.
- Per 17 Oktober 2022, DMS 3.4.7 sekarang mendukung kelas instans Amazon EC2 Generasi 6 untuk instans replikasi.
- Mulai 25 November 2022, dengan DMS 3.4.7 Anda dapat mengonversi skema database dan objek kode menggunakan Konversi Skema DMS, dan menemukan database di lingkungan jaringan Anda yang merupakan kandidat yang baik untuk migrasi menggunakan DMS Fleet Advisor.
- Per 25 November 2022, DMS Studio sudah pensiun.
- Per 31 Januari 2023, Konversi Skema DMS mendukung Aurora MySQL dan Aurora PostgreSQL sebagai penyedia data target.
- Mulai 6 Maret 2023, Anda dapat menghasilkan rekomendasi target berukuran tepat untuk basis data sumber Anda dengan DMS Fleet Advisor.
- Per 6 Maret 2023, AWS DMS mendukung kebijakan AWS terkelola yang memungkinkan penerbitan titik data metrik ke Amazon CloudWatch.

Masalah diselesaikan dalam rilis pemeliharaan DMS 3.4.7 tertanggal 5-Mei-2023

Topik	Penyelesaian
Kegagalan tugas sumber PostgreSQL	Memperbaiki masalah untuk sumber PostgreSQL di mana tugas gagal saat melebihi operasi DDL maksimum yang diizinkan dalam satu peristiwa.
Validasi data sumber PostgreSQL positif palsu	Memperbaiki masalah untuk sumber PostgreSQL dengan target Oracle di mana casting bidang

Topik	Penyelesaian
	stempel waktu yang salah mengakibatkan kesalahan validasi data positif palsu.
Penanganan kesalahan sumber MySQL	Memperbaiki masalah untuk sumber MySQL di mana tugas DMS tidak gagal ketika log BIN berikutnya tidak tersedia.
Pencatatan ROTATE_EVENT sumber MySQL	Peningkatan logging untuk sumber MySQL yang terkait dengan ROTATE_EVENT - termasuk nama log BIN sedang dibaca.
Masalah batas waktu validasi data	Memperbaiki masalah untuk fitur Validasi Data di mana pengaturan <code>executeTimeout</code> titik akhir tidak dihormati untuk kueri yang terkait dengan validasi data.
PostgreSQL menargetkan masalah beban penuh paralel	Memperbaiki masalah untuk target PostgreSQL di mana beban penuh tersegmentasi (paralel) gagal karena kesalahan "koneksi turun".
Masalah pemindahan tugas DMS	Memperbaiki masalah untuk target S3 di mana operasi pemindahan tugas DMS memakan waktu yang sangat lama atau tidak pernah selesai.
Masalah catatan duplikat sumber PostgreSQL	Memperbaiki masalah untuk sumber PostgreSQL di mana tugas DMS akan memunculkan kesalahan yang terkait dengan duplikat pada target setelah tugas berhenti dan dilanjutkan.
Validasi data target Oracle positif palsu	Memperbaiki masalah untuk target Oracle di mana validasi data akan melaporkan kesalahan positif palsu karena zona waktu yang direplikasi salah untuk bidang stempel waktu.

Masalah diselesaikan dalam rilis pemeliharaan DMS 3.4.7 tertanggal 22-Februari-2023

Topik	Penyelesaian
Replika SQL Server AG sebagai sumber	Menambahkan dukungan untuk sumber SQL Server dalam AlwaysOn konfigurasi di mana port TCP pendengar berbeda dari port TCP replika.
Kehilangan data dengan Amazon Redshift sebagai target	Memperbaiki masalah untuk target Redshift di mana dalam beberapa kasus yang jarang terjadi restart Redshift yang tidak terduga dapat menyebabkan data yang hilang pada target.
Dukungan perlindungan sumber SQL Server	Memperbaiki masalah untuk sumber SQL Server di mana tugas DMS bisa gagal dengan kesalahan yang menunjukkan ketidakmampuan untuk membaca cadangan log transaksi saat Pengaturan Titik Akhir ditentukan. "SafeguardPolicy": "EXCLUSIVE_AUTOMATIC_TRUNCATION"
Kegagalan tugas validasi data untuk Oracle sebagai sumber	Memperbaiki masalah untuk sumber Oracle di mana tugas DMS bisa gagal pada validasi data karena nilai Kunci Utama yang salah diidentifikasi.
Kinesis sebelum masalah data gambar	Memperbaiki masalah untuk target streaming (Kinesis, Kafka) di mana pengaturan "EnableBeforeImage" tugas hanya berfungsi untuk tipe data karakter.
File log Perjalanan Waktu	Memperbaiki masalah untuk fitur Perjalanan Waktu di mana DMS membuat file log perjalanan waktu nol byte saat sumbernya mengganggu.

Masalah diselesaikan dalam rilis pemeliharaan DMS 3.4.7 tertanggal 16-Desember-2022

Topik	Penyelesaian
BatchApplyEnabled	Memperbaiki masalah logging yang berlebihan saat BatchApplyEnabled disetel ke True.
Pengaturan titik akhir MongoDB baru— FullLoadNoCursorTimeout	FullLoadNoCursorTimeout Pengaturan titik akhir MongoDB NoCursorTimeout menentukan untuk cursor beban penuh. NoCursorTimeout adalah pengaturan koneksi MongoDB yang mencegah server menutup cursor jika idle.
MongoDB — fungsi filter untuk segmentasi kolom tunggal	Fungsi filter baru meningkatkan kinerja untuk memigrasi database MongoDB menggunakan satu kolom untuk segmentasi.
MongoDB ke Redshift	Saat bermigrasi dari MongoDB ke Redshift, jika koleksi MongoDB memiliki tipe data biner, memperbaiki masalah di mana DMS tidak membuat tabel target di Redshift.
Atribut koneksi SocketTimeout MongoDB MS baru	Atribut koneksi tambahan SocketTimeout MongoDB MS baru mengkonfigurasi batas waktu koneksi untuk klien MongoDB dalam satuan milidetik. Jika nilainya kurang dari atau sama dengan nol, maka default klien MongoDB digunakan.
Memperbaiki masalah yang menyebabkan tugas Amazon Kinesis mogok	Saat bermigrasi ke Amazon Kinesis Data Streams sebagai target, memperbaiki masalah yang menangani nilai null jika kunci utama tidak ada dalam tabel.
Validasi data Oracle NULL PK/UK didukung	Menghapus batasan bahwa validasi data nilai NULL PK/UK tidak didukung.

Topik	Penyelesaian
Oracle ke Amazon S3	Saat bermigrasi dari Oracle ke Amazon S3, memperbaiki masalah di mana beberapa catatan salah dimigrasi sebagai NULL.
Oracle Siaga	Saat menggunakan Oracle Standby sebagai sumber, menambahkan kemampuan DMS untuk menangani transaksi terbuka.
Migrasi Oracle ke Oracle dengan tipe data spasial SDO_GEOMETRY	Saat bermigrasi dari Oracle ke Oracle, memperbaiki masalah di mana tugas gagal jika tabel memiliki kolom SDO_GEOMETRY yang ada di DDL.
Oracle sebagai sumber	Saat menggunakan Oracle sebagai sumber, memperbaiki masalah di mana DMS sesekali melewatkan nomor urutan log redo Oracle.
Oracle sebagai sumber—arsip/log pengulangan online yang hilang	Saat menggunakan Oracle sebagai sumber, memperbaiki masalah sehingga tugas DMS gagal saat log arsip hilang.
Tetap — DMS sesekali melewatkan log pengulangan siaga Oracle	Saat menggunakan Oracle sebagai sumber, memperbaiki masalah di mana DMS sesekali melewatkan nomor urutan log redo Oracle.
Tipe data spasial Oracle ke Oracle yang tidak bereplikasi selama CDC	Saat mereplikasi dari Oracle ke Oracle, memperbaiki masalah di mana tipe data spasial tidak bereplikasi selama CDC.
Oracle sebagai target	Saat menggunakan Oracle sebagai target, memperbaiki masalah di mana target diterapkan gagal dengan kesalahan ORA-01747.

Topik	Penyelesaian
Amazon S3 — Memperbaiki kehilangan data tabel muat ulang	Saat menggunakan Amazon S3 sebagai target, memperbaiki masalah saat operasi muat ulang tabel tidak menghasilkan file CDC.
Tetap — SQL Server Selalu Pada inisialis asi konteks dalam kasus server utama sebagai sumber	Saat menggunakan SQL Server Always On sebagai sumber, memperbaiki masalah untuk tidak menginisialisasi Grup Ketersediaan (AG) jika sumbernya primer dan AlwaysOnSharedSyncedBackupsEnabled disetel ke true.
Pengaturan titik akhir SQL Server yang diperbarui	Ketika titik akhir sumber adalah SQL Server Always On Availability Group dan merupakan replika sekunder, memperbaiki masalah saat tugas replikasi gagal jika AlwaysOnSharedSyncedBackupsEnabled disetel ke True.
PostgreSQL sebagai sumber	Memperbaiki masalah saat CDC gagal memigrasi operasi hapus/pembaruan pada sumber PostgreSQL, yang diperkenalkan di 3.4.7 dalam mendukung Boolean. mapBooleanAs

AWS Catatan rilis Database Migration Service 3.4.6

Tabel berikut menunjukkan fitur dan penyempurnaan baru yang diperkenalkan di AWS Database Migration Service (AWS DMS) versi 3.4.6.

Fitur baru atau perangkat tambahan	Deskripsi
AWS DMS Perjalanan Waktu	AWS DMS memperkenalkan Time Travel , fitur yang memberikan fleksibilitas kepada pelanggan pada kemampuan logging mereka, dan meningkatkan pengalaman pemecahan masalah mereka. Dengan Time Travel, Anda dapat menyimpan dan mengenkripsi

Fitur baru atau perangkat tambahan	Deskripsi
	<p>AWS DMS log menggunakan Amazon S3, dan melihat, mengunduh, dan mengagurkan log dalam jangka waktu tertentu.</p>
Mendukung Microsoft Azure SQL Managed Instance sebagai sumber	<p>AWS DMS sekarang mendukung Microsoft Azure SQL Managed Instance sebagai sumber. Dengan menggunakan AWS DMS, Anda sekarang dapat melakukan migrasi langsung dari Microsoft Azure SQL Managed Instance ke target yang AWS DMS didukung.</p> <p>Untuk informasi tentang AWS DMS sumber, lihat Sumber untuk migrasi data.</p> <p>Untuk informasi tentang AWS DMS target yang didukung, lihat Target migrasi data.</p>
Mendukung Google Cloud SQL untuk MySQL sebagai sumber	<p>AWS DMS sekarang mendukung Google Cloud SQL untuk MySQL sebagai sumber. Dengan menggunakan AWS DMS, Anda sekarang dapat melakukan migrasi langsung dari Google Cloud SQL untuk MySQL ke target yang didukung. AWS DMS</p> <p>Untuk informasi tentang AWS DMS sumber, lihat Sumber untuk migrasi data.</p> <p>Untuk informasi tentang AWS DMS target yang didukung, lihat Target migrasi data.</p>
Support parallel load untuk data yang dipartisi ke S3	<p>AWS DMS sekarang mendukung beban paralel untuk data yang dipartisi ke Amazon S3, meningkatkan waktu muat untuk memigrasi data yang dipartisi dari data sumber mesin database yang didukung ke Amazon S3. Fitur ini membuat sub-folder Amazon S3 untuk setiap partisi tabel di sumber database, memungkinkan untuk AWS DMS menjalankan proses paralel untuk mengisi setiap sub-folder.</p>

Fitur baru atau perangkat tambahan	Deskripsi
Mendukung beberapa topik target Apache Kafka dalam satu tugas	AWS DMS sekarang mendukung target multi-topik Apache Kafka dengan satu tugas. Dengan menggunakan AWS DMS, Anda sekarang dapat mereplikasi beberapa skema dari satu database ke topik target Apache Kafka yang berbeda menggunakan tugas yang sama. Ini menghilangkan kebutuhan untuk membuat beberapa tugas terpisah dalam situasi di mana banyak tabel dari database sumber yang sama perlu dimigrasikan ke topik target Kafka yang berbeda.

Masalah yang diselesaikan di AWS DMS 3.4.6 meliputi yang berikut:

- Memperbaiki masalah saat kolom dari pernyataan UPDATE diisi ke kolom yang salah jika kolom kunci utama bukan kolom pertama saat menggunakan Amazon S3 sebagai target dengan format CSV.
- Memperbaiki masalah saat AWS DMS tugas mungkin macet saat menggunakan plugin pglogical dengan NULL nilai dalam BYTEA kolom di bawah mode LOB terbatas saat menggunakan PostgreSQL sebagai sumber.
- Memperbaiki masalah saat AWS DMS tugas mungkin macet saat sejumlah besar tabel sumber dihapus saat menggunakan PostgreSQL sebagai sumber.
- Peningkatan partisi folder berbasis tanggal Amazon S3 dengan memperkenalkan setelan Amazon S3 baru untuk memungkinkan partisi pada tanggal non-UTC. `DatePartitionTimezone`
- Mendukung pemetaan antara tipe data `TIMESTAMP WITH TIME ZONE` dari sumber hingga `TIMESTAMPTZ` saat menggunakan Redshift sebagai target
- Meningkatkan kinerja CDC untuk tugas tanpa aturan pemilihan wildcard saat menggunakan MongoDB atau Amazon DocumentDB sebagai sumber.
- Memperbaiki masalah di mana nama skema dengan wildcard garis bawah dan panjang kurang dari 8 tidak ditangkap oleh AWS DMS tugas saat menggunakan Db2 LUW sebagai sumber.
- Memperbaiki masalah AWS DMS saat instance kehabisan memori di bawah volume data yang besar saat menggunakan OpenSearch Service sebagai target.
- Meningkatkan kinerja validasi data dengan mendukung tugas validasi full load saja.

- Memperbaiki masalah saat AWS DMS tugas gagal dilanjutkan setelah failover paksa saat menggunakan Sybase sebagai sumber.
- Memperbaiki masalah saat AWS DMS mengirim peringatan Invalid BC timestamp was encountered in column secara tidak benar.

Masalah yang diselesaikan dalam rilis pemeliharaan DMS 3.4.6 meliputi:

- Memperbaiki masalah tugas mogok saat mode penerapan massal diaktifkan saat menggunakan Oracle sebagai sumber dan target.
- Memperbaiki masalah sehingga tugas pemuatan penuh menggunakan pengaturan `ExecuteTimeout` titik akhir dengan PostgreSQL sebagai sumber dengan benar.
- Memperbaiki masalah dengan memigrasi kolom tipe data Array saat tugas diatur ke mode LOB terbatas saat menggunakan PostgreSQL sebagai sumber.
- Memperbaiki masalah migrasi stempel waktu dengan zona waktu sebelum 1970-01-01 saat menggunakan PostgreSQL sebagai sumber.
- Memperbaiki masalah saat DMS memperlakukan string kosong sebagai null selama replikasi saat menggunakan SQL Server sebagai sumber dan target.
- Memperbaiki masalah untuk menghormati pengaturan titik akhir baca dan tulis sesi tunggu saat menggunakan sumber/target MySQL.
- Memperbaiki masalah saat tugas DMS CDC mengunduh file terkait beban penuh saat menggunakan Amazon S3 sebagai sumber.
- Memperbaiki masalah log crash saat `CdcInsertsAndUpdates` dan keduanya `PreserveTransactions` disetel `true` saat menggunakan Amazon S3 sebagai target.
- Memperbaiki masalah saat tugas macet saat fitur `ParallelApply` * diaktifkan, tetapi beberapa tabel tidak memiliki kunci primer default saat menggunakan Amazon Kinesis Data Streams sebagai sumber.
- Memperbaiki masalah saat kesalahan tidak diberikan karena kesalahan `StreamArn` saat menggunakan Amazon Kinesis Data Streams sebagai sumber.
- Memperbaiki masalah di mana nilai kunci utama sebagai string kosong akan menyebabkan tugas macet saat digunakan OpenSearch sebagai target.
- Memperbaiki masalah di mana terlalu banyak ruang disk digunakan oleh validasi data.

Masalah diselesaikan dalam rilis pemeliharaan DMS 3.4.6 tertanggal 13-Desember-2022

Topik	Penyelesaian
Pengemudi SAP ASE odbcc	Memperbaiki masalah untuk SAP ASE sebagai sumber sehingga driver ODBC dapat mendukung set karakter.
Bug kunci utama datatype sqlserver untuk pencarian lob	Memperbaiki masalah untuk SQL Server sebagai sumber di mana pencarian LOB tidak berfungsi dengan baik, ketika kunci utama memiliki tipe data datetime, dengan presisi dalam milidetik.
SQL Server ke RedShift — 'DateTime Offset' dipetakan ke 'timestampz'	Untuk migrasi dari SQL Server ke Redshift, pemetaan yang ditingkatkan sehingga format 'datetimeoffset' SQL Server dipetakan ke format 'timestampz' Redshift.
Validasi Data - SkipLobColumns Benar	Memperbaiki masalah saat tugas DMS mogok saat SkipLobColumns True, ada LOB pada sumbernya, Kunci Utama ada di kolom terakhir, dan perbedaan data terdeteksi oleh validasi.
Validasi Data dengan MySQL sebagai sumber	Memperbaiki masalah untuk MySQL sebagai sumber dengan validasi data diaktifkan, di mana kerusakan tugas DMS terjadi saat menggunakan tabel yang memiliki kunci unik komposit yang memiliki nilai nol.
MySQL sebagai sumber	Memperbaiki masalah untuk MySQL sebagai sumber, di mana tabel ditangguhkan dengan kesalahan Overflow saat kolom diubah untuk menambah presisi.
Tingkatkan driver MySQL ODBC ke 8.0.23	Memperbaiki masalah untuk MySQL sebagai sumber, di mana pengumpulan 'utf8mb4_0900_bin' tidak kompatibel dengan driver mysql yang digunakan oleh DMS.
MySQL — Mendukung perubahan DDL untuk tabel yang dipartisi	Memperkenalkan skipTableSuspension ForPartitionDdl pengaturan endpoint MySQL baru untuk memungkinkan pengguna melewati suspensi tabel untuk perubahan DDL partisi selama CDC, sehingga DMS sekarang

Topik	Penyelesaian	
	dapat mendukung perubahan DDL untuk tabel MySQL yang dipartisi.	
Migrasi MongoDB ke Redshift	Memperbaiki masalah migrasi MongoDB ke Redshift, di mana DMS gagal membuat tabel target di Redshift jika koleksi MongoDB memiliki tipe data biner.	
Target Redshift — Segfault Perjalanan Waktu dalam Massal Terapkan	Memperbaiki masalah untuk Redshift sebagai target, di mana tugas DMS mogok dengan BatchApplyEnabled disetel ke true.	
Redshift sebagai target	Memperbaiki masalah untuk Redshift sebagai target, di mana dengan beban paralel disetel ke type=partitions-auto, segmen paralel menulis file CSV massal ke direktori tabel yang sama dan saling mengganggu.	
Redshift sebagai target	Memperbaiki masalah dengan Redshift sebagai target, di mana selama CDC kolom target bertipe boolean sementara sumbernya bertipe karakter bervariasi.	
Redshift sebagai target	Memperbaiki log tugas untuk mengidentifikasi perubahan DDL yang gagal direplikasi ke Redshift sebagai target.	
Validasi data dengan PostgreSQL	Memperbaiki masalah validasi dengan PostgreSQL, di mana validasi gagal saat tipe data boolean hadir.	
PostgreSQL sebagai sumber	Memperbaiki masalah untuk PostgreSQL sebagai sumber, sehingga beban penuh menggunakan ExecuteTimeout bidang dalam atribut koneksi Ekstra.	

Topik	Penyelesaian
PostgreSQL sebagai sumber	Memperbaiki masalah untuk PostgreSQL sebagai sumber, sehingga tugas akan gagal jika membaca LSN yang lebih besar dari tugas yang diminta melanjutkan LSN selama lebih dari 60 menit untuk menunjukkan bahwa itu adalah masalah dengan slot replikasi yang digunakan.
PostgreSQL sebagai sumber—timestampz sebelum 1970-01-01	Memperbaiki masalah untuk PostgreSQL sebagai sumber, di mana timestampz sebelum 1970-01-01 tidak dimigrasi dengan benar selama CDC.
PostgreSQL sebagai sumber	Memperbaiki masalah untuk PostgreSQL sebagai sumber, di mana DMS memotong karakter berbagai nilai tipe data selama CDC.
PostgreSQL sebagai sumber—melanjutkan tugas yang dihentikan	Memperbaiki masalah untuk PostgreSQL sebagai sumber di mana melanjutkan pemutaran ulang tugas yang sebelumnya dihentikan melewati satu atau beberapa transaksi selama CDC.
Amazon S3 sebagai target	Memperbaiki masalah untuk S3 sebagai target, di mana header file CSV yang dihasilkan dimatikan oleh satu kolom saat AddColumnName benar dan Timestamp ColumnName "".
Amazon S3 sebagai perilaku penggunaan sumber-memori dalam fase beban penuh untuk tugas	Memperbaiki masalah untuk S3 sebagai sumber, di mana tugas DMS dalam beban penuh hanya melepaskan memori yang digunakan setelah seluruh tabel dimuat ke database target.
Amazon S3 sebagai operasi muat ulang tabel target	Memperbaiki masalah untuk S3 sebagai target, di mana operasi muat ulang tabel tidak menghasilkan file CDC.

AWS Catatan rilis Database Migration Service 3.4.5

Tabel berikut menunjukkan fitur dan penyempurnaan baru yang diperkenalkan di AWS Database Migration Service (AWS DMS) versi 3.4.5.

Fitur baru atau perangkat tambahan	Deskripsi
Support untuk Redis sebagai target	AWS DMS sekarang mendukung Redis sebagai target. Dengan menggunakan AWS DMS, Anda sekarang dapat memigrasikan data langsung dari sumber apa pun yang AWS DMS didukung ke penyimpanan data Redis, dengan waktu henti minimal. Untuk informasi tentang AWS DMS target, lihat Target migrasi data .
Support untuk MongoDB 4.2 dan 4.4 sebagai sumber	AWS DMS sekarang mendukung MongoDB 4.2 dan 4.4 sebagai sumber. Dengan menggunakan AWS DMS, Anda sekarang dapat memigrasikan data dari MongoDB 4.2 dan 4.4 cluster ke AWS DMS target yang didukung termasuk Amazon DocumentDB (dengan kompatibilitas MongoDB), dengan downtime minimal. Untuk informasi tentang AWS DMS sumber, lihat Sumber untuk migrasi data .
Support untuk beberapa database menggunakan MongoDB sebagai sumber	AWS DMS sekarang mendukung migrasi beberapa database dalam satu tugas menggunakan MongoDB sebagai sumber. Dengan menggunakan AWS DMS, Anda sekarang dapat mengelompokkan beberapa database kluster MongoDB, dan memigrasikannya menggunakan satu tugas migrasi database. Anda dapat bermigrasi ke target yang AWS DMS didukung, termasuk Amazon DocumentDB (dengan kompatibilitas MongoDB), dengan waktu henti minimal.
Support untuk segmentasi otomatis menggunakan MongoDB atau Amazon DocumentDB (dengan kompatibilitas MongoDB) sebagai sumber	AWS DMS sekarang mendukung segmentasi otomatis menggunakan MongoDB atau Amazon DocumentDB sebagai sumber. Dengan menggunakan AWS DMS, Anda dapat mengonfigurasi tugas migrasi database untuk menyegmentasikan koleksi cluster MongoDB atau DocumentDB secara otomatis. Anda kemudian dapat memigrasikan segmen secara paralel ke target yang AWS

Fitur baru atau perangkat tambahan	Deskripsi
	DMS didukung, termasuk Amazon DocumentDB, dengan waktu henti minimal.
Peningkatan kinerja beban penuh Amazon Redshift	AWS DMS sekarang mendukung penggunaan thread paralel saat menggunakan Amazon Redshift sebagai target selama pemuatan penuh. Dengan memanfaatkan pengaturan tugas pemuatan penuh multithreaded, Anda dapat meningkatkan kinerja migrasi awal dari sumber apa pun yang AWS DMS didukung ke Amazon Redshift. Untuk informasi tentang AWS DMS target, lihat Target migrasi data .

Masalah yang diselesaikan di AWS DMS 3.4.5 meliputi yang berikut:

- Memperbaiki masalah di mana data berpotensi hilang atau digandakan setelah dilanjutkan, saat menggunakan PostgreSQL sebagai sumber dengan konkurensi transaksi tinggi.
- Memperbaiki masalah saat tugas migrasi database gagal dengan kesalahan Tidak dapat menemukan id relasi... saat menggunakan PostgreSQL sebagai sumber, dengan plugin pglogical diaktifkan.
- Memperbaiki masalah di mana VARCHAR kolom tidak direplikasi dengan benar saat menggunakan PostgreSQL sebagai sumber dan Oracle sebagai target.
- Memperbaiki masalah saat operasi penghapusan tidak ditangkap dengan benar saat kunci utama bukan kolom pertama dalam definisi tabel, saat menggunakan PostgreSQL sebagai sumber.
- Memperbaiki masalah saat tugas migrasi database melewati pembaruan LOB dalam pengaturan metadata khusus saat menggunakan MySQL sebagai sumber.
- Memperbaiki masalah di mana TIMESTAMP kolom diperlakukan seperti DATETIME dalam mode LOB penuh saat menggunakan MySQL versi 8 sebagai sumber.
- Memperbaiki masalah saat tugas migrasi database gagal saat mengurai NULL DATETIME catatan saat menggunakan MySQL 5.6.4 dan yang lebih tinggi sebagai sumber.
- Memperbaiki masalah saat tugas migrasi database macet setelah menemukan kesalahan Thread keluar saat menggunakan Amazon Redshift sebagai target dengan penerapan paralel.
- Memperbaiki masalah saat data berpotensi hilang, saat tugas migrasi database terputus dengan titik akhir target Amazon Redshift selama CDC penerapan batch.

- Meningkatkan kinerja beban penuh dengan menelepon `ACCEPTINVCHARS` saat menggunakan Amazon Redshift sebagai target.
- Memperbaiki masalah saat rekaman duplikat direplikasi saat kembali dari mode one-by-one ke mode penerapan paralel menggunakan Amazon Redshift sebagai target.
- Memperbaiki masalah saat tugas migrasi database tidak mengalihkan kepemilikan objek Amazon S3 ke pemilik bucket `cannedAclForObjects=bucket_owner_full_control` saat menggunakan Amazon S3 sebagai target.
- Ditingkatkan AWS DMS dengan mendukung beberapa tujuan arsip dengan `ECA additionalArchivedLogDestId` saat menggunakan Oracle sebagai sumber.
- Memperbaiki masalah saat tugas migrasi database gagal dengan kesalahan `OCI_INVALID_HANDLE` saat memperbarui kolom LOB dalam mode LOB penuh.
- Memperbaiki masalah saat `NVARCHAR2` kolom tidak dimigrasi dengan benar selama CDC saat menggunakan Oracle sebagai sumber.
- Ditingkatkan AWS DMS dengan mengaktifkan `SafeguardPolicy` saat menggunakan RDS untuk SQL Server sebagai sumber.
- Memperbaiki masalah saat tugas migrasi database melaporkan kesalahan `rdsadmin` saat menggunakan sumber SQL Server non-RDS.
- Memperbaiki masalah di mana validasi data gagal dengan `UUID` sebagai kunci utama dalam pengaturan partisi saat menggunakan SQL Server sebagai sumber.
- Memperbaiki masalah di mana beban penuh ditambah tugas CDC mungkin gagal jika `LSN` yang diperlukan tidak dapat ditemukan di log database saat menggunakan `Db2 LUW` sebagai sumber.
- Ditingkatkan AWS DMS dengan mendukung stempel waktu CDC kustom saat menggunakan MongoDB sebagai sumber.
- Memperbaiki masalah di mana tugas migrasi database macet saat berhenti, menggunakan MongoDB sebagai sumber, ketika kesalahan driver MongoDB aktif. `endSessions`
- Memperbaiki masalah saat AWS DMS gagal memperbarui bidang non-primer saat menggunakan DynamoDB sebagai target
- Memperbaiki masalah saat validasi data melaporkan ketidakcocokan positif palsu pada `CLOB` dan `NCLOB` kolom.
- Memperbaiki masalah saat validasi data gagal pada catatan khusus spasi saat menggunakan Oracle sebagai sumber.
- Memperbaiki masalah saat tugas migrasi database mogok saat memotong tabel yang dipartisi.

- Memperbaiki masalah saat tugas migrasi database gagal saat membuat tabel `awsdms_apply_exceptions` kontrol.
- Dukungan diperpanjang dari plugin `caching_sha2_password` otentikasi saat menggunakan MySQL versi 8.

AWS Catatan rilis Database Migration Service 3.4.4

Tabel berikut menunjukkan fitur baru dan memperkenalkan perangkat tambahan di AWS DMS versi 3.4.4.

Fitur baru atau perangkat tambahan	Deskripsi
Enkripsi Support TLS dan autentikasi TLS atau SASL menggunakan Kafka sebagai target	AWS DMS sekarang mendukung enkripsi TLS dan otentikasi TLS atau SASL menggunakan Amazon MSK dan kluster Kafka lokal sebagai target. Untuk informasi lebih lanjut tentang penggunaan enkripsi dan autentikasi untuk titik akhir Kafka, lihat Menghubungkan ke Kafka menggunakan Keamanan Lapisan Pengangkutan (TLS) .

Masalah yang diselesaikan di AWS DMS 3.4.4 meliputi yang berikut:

- Peningkatan AWS DMS logging pada kegagalan tugas saat menggunakan endpoint Oracle.
- Eksekusi AWS DMS tugas yang ditingkatkan terus diproses ketika titik akhir sumber Oracle beralih peran setelah Oracle Data Guard gagal selesai.
- Penanganan perbaikan kesalahan memperlakukan ORA—12561 sebagai kesalahan yang dapat dipulihkan saat menggunakan titik akhir Oracle.
- Memperbaiki masalah di mana kolom `EMPTY_BLOB()` dan `EMPTY_CLOB()` mengalami migrasi yang tidak berlaku ketika menggunakan Oracle sebagai sumber.
- Memperbaiki masalah saat AWS DMS tugas gagal memperbarui catatan setelah menambahkan perubahan kolom DDL saat menggunakan SQL Server sebagai sumber.
- Perbaikan PostgreSQL sebagai migrasi sumber dengan support tipe data `TIMESTAMP WITH TIME ZONE`.

- Memperbaiki masalah di mana pengaturan `afterConnectScript` tidak bekerja selama beban penuh ketika menggunakan PostgreSQL sebagai target.
- Memperkenalkan pengaturan `mapUnboundedNumericAsString` yang baru agar lebih baik dalam menangani tipe tanggal NUMERIC tanpa presisi dan menskalakan saat menggunakan titik akhir PostgreSQL.
- Memperbaiki masalah saat AWS DMS tugas gagal dengan “0 baris terpengaruh” setelah menghentikan dan melanjutkan tugas saat menggunakan PostgreSQL sebagai sumber.
- Memperbaiki masalah saat AWS DMS gagal memigrasikan tipe TIMESTAMP data dengan BC akhiran saat menggunakan PostgreSQL sebagai sumber.
- Memperbaiki masalah saat AWS DMS gagal memigrasikan TIMESTAMP nilai “±infinity” saat menggunakan PostgreSQL sebagai sumber.
- Memperbaiki masalah di mana rangkaian kosong diperlakukan sebagai NULL saat menggunakan S3 sebagai sumber dengan pengaturan `csvNullValue` yang diatur ke nilai-nilai lain.
- Peningkatan `timestampColumnName` atribut koneksi tambahan dalam beban penuh dengan CDC harus diurutkan selama CDC ketika menggunakan S3 sebagai target.
- Peningkatan penanganan tipe data biner dalam format hex seperti BYTE, BINARY, dan BLOB saat menggunakan S3 sebagai sumber.
- Memperbaiki masalah saat data yang dihapus dilakukan migrasi dengan karakter khusus saat menggunakan S3 sebagai target.
- Memperbaiki masalah untuk menangani nilai kunci kosong saat menggunakan Amazon DocumentDB (dengan kompatibilitas MongoDB) sebagai target.
- Memperbaiki masalah saat AWS DMS gagal mereplikasi `NumberDecimal` atau `Decimal128` kolom saat menggunakan MongoDB atau Amazon DocumentDB (dengan kompatibilitas MongoDB) sebagai sumber.
- Memperbaiki masalah untuk mengizinkan tugas CDC untuk mencoba lagi ketika gagal pada MongoDB atau Amazon DocumentDB (dengan kompatibilitas MongoDB) sebagai sumber.
- Menambahkan opsi untuk menghapus awalan heksadesimal “0x” ke nilai RAW tipe data saat menggunakan Kinesis, Kafka, atau sebagai target. OpenSearch
- Memperbaiki masalah di mana validasi gagal pada kolom karakter panjang tetap saat menggunakan Db2 LUW sebagai sumber.
- Memperbaiki masalah saat validasi gagal bila hanya tipe data sumber atau tipe data target FLOAT atau DOUBLE.

- Memperbaiki masalah di mana validasi gagal pada karakter NULL saat menggunakan Oracle sebagai sumber.
- Memperbaiki masalah di mana validasi gagal pada kolom XML saat menggunakan Oracle sebagai sumber.
- Memperbaiki masalah saat AWS DMS tugas macet saat ada kolom nullable di kunci komposit menggunakan MySQL sebagai sumber.
- Memperbaiki masalah saat AWS DMS gagal memvalidasi kedua UNIQUEIDENTIFIER kolom dari titik akhir sumber SQL Server dan kolom UUID dari titik akhir target PostgreSQL.
- Memperbaiki masalah di mana tugas CDC tidak menggunakan definisi tabel sumber yang diperbarui setelah diubah.
- Peningkatan AWS DMS gagal untuk menangani kegagalan tugas yang disebabkan oleh nama pengguna atau kata sandi yang tidak valid sebagai kesalahan yang dapat dipulihkan.
- Memperbaiki masalah saat AWS DMS tugas gagal karena LSNs yang hilang saat menggunakan RDS untuk SQL Server sebagai sumber.

AWS Catatan rilis Database Migration Service 3.4.3

Tabel berikut menunjukkan fitur baru dan memperkenalkan perangkat tambahan di AWS DMS versi 3.4.3.

Fitur baru atau perangkat tambahan	Deskripsi
Versi Amazon DocumentDB baru	Amazon DocumentDB versi 4.0 sekarang didukung sebagai sumber.
Versi MariaDB baru	MariaDB versi 10.4 sekarang didukung sebagai sumber dan target.
Support untuk AWS Secrets Manager integrasi	Anda dapat menyimpan detail koneksi basis data (kredensial pengguna) untuk titik akhir yang didukung dengan aman di AWS Secrets Manager. Anda kemudian dapat mengirimkan rahasia yang sesuai alih-alih kredensial teks biasa AWS DMS saat Anda membuat atau memodifikasi titik akhir. AWS DMS kemudian terhubung ke database endpoint menggunakan rahasia. Untuk informasi selengkapnya tentang membuat rahasia untuk AWS DMS

Fitur baru atau perangkat tambahan	Deskripsi
	titik akhir, lihat Menggunakan rahasia untuk mengakses titik akhir AWS Database Migration Service .
Pilihan yang lebih besar untuk instans replikasi C5 dan R5	Anda sekarang dapat membuat pengukuran instans replikasi yang lebih besar berikut: pengukuran C5 hingga 96 vCPUs dan 192 GiB memori dan pengukuran R5 hingga 96 vCPUs dan 768 GiB memori.
Peningkatan performa Amazon Redshift	AWS DMS sekarang mendukung penerapan paralel saat menggunakan Redshift sebagai target untuk meningkatkan kinerja replikasi yang sedang berlangsung. Untuk informasi selengkapnya, lihat Pengaturan tugas multithreaded untuk Amazon Redshift .

Masalah yang diselesaikan di AWS DMS 3.4.3 meliputi yang berikut:

- Memperbaiki masalah dengan melakukan stempel waktu menjadi “1970-01-01 00:00:00” untuk tindakan yang ditangguhkan saat menggunakan Db2 LUW sebagai sumber.
- Memperbaiki masalah saat AWS DMS tugas gagal dengan NVARCHAR kolom sebagai kunci utama saat menggunakan SQL Server sebagai sumber dengan mode LOB Penuh.
- Memperbaiki masalah catatan yang hilang selama fase perubahan cache saat menggunakan SQL Server sebagai sumber.
- Memperbaiki masalah saat catatan dilewati setelah AWS DMS tugas dilanjutkan saat menggunakan RDS untuk SQL Server sebagai sumber.
- Memperbaiki masalah di mana komponen logging AWS DMS ASSERTION menghasilkan log besar untuk SQL Server.
- Memperbaiki masalah di mana validasi data gagal selama fase CDC karena penguraian kolom melimpah saat menggunakan MySQL sebagai sumber.
- Memperbaiki masalah saat AWS DMS tugas macet karena kesalahan segmentasi selama validasi data saat menggunakan PostgreSQL sebagai target.
- Memperbaiki masalah di mana validasi data gagal pada tipe data DOUBLE selama CDC saat menggunakan PostgreSQL sebagai sumber dan target.

- Memperbaiki masalah di mana catatan yang dimasukkan oleh perintah salinan tidak dilakukan replikasi dengan benar saat menggunakan PostgreSQL sebagai sumber dan Redshift sebagai target.
- Memperbaiki masalah kehilangan data selama fase perubahan cache saat menggunakan PostgreSQL sebagai sumber.
- Memperbaiki masalah yang berpotensi menyebabkan kehilangan data atau duplikat catatan saat menggunakan PostgreSQL sebagai sumber.
- Memperbaiki masalah di mana skema dengan berbagai campuran kasus gagal melakukan migrasi dengan pglogical saat menggunakan PostgreSQL sebagai sumber.
- Memperbaiki masalah di mana Pesan Gagal Terakhir tidak berisi kesalahan ORA saat menggunakan Oracle sebagai sumber.
- Memperbaiki masalah saat AWS DMS tugas gagal membuat pernyataan UPDATE saat menggunakan Oracle sebagai target.
- Memperbaiki masalah saat AWS DMS tugas tidak mereplikasi data saat menggunakan Oracle 12.2 sebagai sumber dengan konfigurasi ASM dan Pluggable Database.
- Perbaiki catatan penguraian dengan mempertahankan kutipan agar patuh dengan RFC 4180 saat menggunakan S3 sebagai sumber.
- Perbaiki penanganan `timestampColumnName` sehingga kolom dari beban penuh dapat diurutkan dengan kolom dari CDC.
- Dengan memperkenalkan pengaturan titik akhir baru `messageMaxBytes`, memperbaiki masalah di mana AWS DMS tugas gagal ketika ada elemen LOB yang lebih besar dari 1MB.
- Memperbaiki masalah saat AWS DMS tugas macet karena kesalahan segmentasi saat menggunakan Redshift sebagai target.
- Peningkatan kesalahan dalam mencatat koneksi uji Redshift.
- Memperbaiki masalah di mana AWS DMS tidak mentransfer semua dokumen dari MongoDB ke DocumentDB selama Full Load.
- Memperbaiki masalah saat AWS DMS tugas melaporkan kesalahan fatal saat tidak ada tabel yang disertakan dalam aturan pemetaan tabel.
- Memperbaiki masalah saat skema dan tabel yang dibuat sebelum memulai ulang tugas AWS DMS tidak melakukan replikasi pada target ketika menggunakan MySQL sebagai sumber.
- Memperbaiki masalah di mana wildcard escape `[_]` tidak dapat menghindari wildcard `"_"` dalam aturan pengecualian saat menggunakan MySQL sebagai sumber.

- Memperbaiki masalah di mana kolom tipe data UNSIGNED BIGINT tidak melakukan replikasi dengan benar saat menggunakan MySQL sebagai sumber.

AWS Catatan rilis Database Migration Service 3.4.2

Tabel berikut menunjukkan fitur baru dan memperkenalkan perangkat tambahan di AWS DMS versi 3.4.2.

Fitur baru atau perangkat tambahan	Deskripsi
<p>Dukungan untuk menghubungkan Amazon Virtual Private Cloud (Amazon VPC) secara pribadi AWS ke Database Migration Service (DMS) tanpa memerlukan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect</p>	<p>Anda sekarang dapat terhubung dan mengakses AWS DMS dari VPC Amazon Anda melalui titik akhir antarmuka VPC yang Anda buat. Endpoint antarmuka ini memungkinkan Anda untuk mengisolasi semua aktivitas jaringan instance AWS DMS replikasi Anda dalam infrastruktur jaringan Amazon. Dengan menyertakan referensi ke titik akhir antarmuka ini di semua panggilan API untuk AWS DMS menggunakan AWS CLI atau SDK, Anda memastikan bahwa semua AWS DMS aktivitas tetap tidak terlihat oleh Internet publik. Untuk informasi selengkapnya, lihat Keamanan infrastruktur dalam AWS Database Migration Service.</p> <div data-bbox="544 1186 1507 1402" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Fitur ini tersedia menggunakan semua versi AWS DMS mesin yang didukung.</p> </div>
<p>Partisi folder berbasis tanggal CDC menggunakan Amazon S3 sebagai target</p>	<p>AWS DMS sekarang mendukung partisi folder berbasis tanggal saat mereplikasi data menggunakan S3 sebagai target. Untuk informasi selengkapnya, lihat Menggunakan partisi folder berdasarkan tanggal.</p>

Masalah yang diselesaikan di AWS DMS 3.4.2 meliputi yang berikut:

- Menambahkan STATUPDATE saat melakukan performa migrasi menggunakan Redshift sebagai target.

- Perbaiki tugas validasi dengan memperkenalkan pengaturan baru. `ValidQueryCdcDelaySecond` menunda kueri validasi pertama pada titik akhir sumber dan target untuk membantu meredam pertentangan sumber daya ketika latensi migrasi sedang tinggi.
- Memperbaiki masalah yang AWS DMS membutuhkan waktu lama untuk memulai tugas validasi.
- Memperbaiki masalah di mana catatan kosong yang dihasilkan saat mulai atau pemfilteran stopword tugas replikasi menggunakan S3 sebagai target.
- Memperbaiki masalah saat tugas macet setelah beban penuh selesai.
- Memperbaiki masalah di mana tugas macet saat tabel sumber mengalami kesalahan data saat menggunakan S3 sebagai sumber.
- Memperbaiki masalah di mana tugas macet saat mulai ketika akun pengguna titik akhir sumber dinonaktifkan.
- Memperbaiki masalah di mana tugas gagal saat menggunakan PostgreSQL sebagai sumber dengan `REPLICA IDENTITY FULL`.
- Memperbaiki masalah di mana tugas melewati transaksi saat menggunakan PostgreSQL sebagai sumber dengan plugin `pglogical`.
- Memperbaiki masalah saat AWS DMS tidak menghapus file sumber terkompresi saat menggunakan Redshift sebagai target.
- Memperbaiki masalah di mana tugas validasi melaporkan informasi negatif SALAH saat menggunakan MySQL sebagai sumber dan target dengan tipe data `BIGINT UNSIGNED`.
- Memperbaiki masalah di mana tugas validasi melaporkan informasi positif SALAH saat menggunakan SQL Server sebagai sumber dengan kolom kunci utama sebagai tipe `CHAR`.
- Memperbaiki masalah yang AWS DMS tidak menghapus objek target saat menggunakan `start-replication` untuk memulai tugas replikasi menggunakan S3 sebagai target.
- Memperbaiki beberapa masalah pada validasi data saat menggunakan Db2 sebagai sumber.
- Memperbaiki masalah di mana tugas validasi macet saat menggunakan SQL Server sebagai sumber dengan kolom `VARCHAR` sebagai kunci utama.
- Menambahkan support untuk tipe data `STEMPEL WAKTU DENGAN ZONA WAKTU` saat menggunakan PostgreSQL sebagai sumber

AWS Database Migration Service 3.4.1 Catatan rilis beta

Tabel berikut menunjukkan fitur baru dan memperkenalkan perangkat tambahan di AWS DMS versi 3.4.1 Beta.

Fitur baru atau perangkat tambahan	Deskripsi
Versi MongoDB baru	MongoDB versi 4.0 sekarang didukung sebagai sumber.
Support TLS 1.2 untuk SQL Server	AWS DMS sekarang mendukung TLS 1.2 untuk titik akhir SQL Server.

Masalah yang diselesaikan di AWS DMS 3.4.1 Beta meliputi yang berikut:

- Peningkatan support TDE Oracle 19c.
- Peningkatan support set karakter utf8mb4 dan identitas tipe data menggunakan Redshift sebagai target.
- Peningkatan penanganan gagal tugas replikasi saat menggunakan MySQL karena tidak mencatat sumber dan biner.
- Peningkatan support validasi data pada berbagai tipe data dan set karakter.
- Peningkatan penanganan nilai nihil dengan pengaturan titik akhir baru IncludeNullAndEmpty saat menggunakan Kinesis dan Kafka sebagai target.
- Peningkatan kesalahan dalam mencatat dan menangani saat menggunakan Kafka sebagai target.
- Peningkatan waktu DST offset saat menggunakan SQL Server sebagai sumber.
- Memperbaiki masalah di mana tugas replikasi mencoba membuat tabel yang ada untuk Oracle sebagai target.
- Memperbaiki masalah di mana tugas replikasi macet setelah koneksi basis data mati saat menggunakan Oracle sebagai sumber.
- Memperbaiki masalah di mana tugas replikasi gagal mendeteksi dan menghubungkan kembali ke primer baru saat menggunakan SQL Server sebagai sumber dengan pengaturan AlwaysOn.
- Memperbaiki masalah di mana tugas replikasi tidak menambahkan "D" untuk kolom "OP" dalam syarat-syarat tertentu untuk S3 sebagai target.

AWS Database Migration Service 3.4.0 Catatan rilis beta

Tabel berikut menunjukkan fitur baru dan memperkenalkan perangkat tambahan dalam versi 3.4.0 AWS DMS

Fitur baru atau perangkat tambahan	Deskripsi
Versi MySQL baru	AWS DMS sekarang mendukung MySQL versi 8.0 sebagai sumber, kecuali ketika muatan transaksi dikompresi.
Support TLS 1.2 untuk MySQL	AWS DMS sekarang mendukung TLS 1.2 untuk endpoint MySQL.
Versi MariaDB baru	AWS DMS sekarang mendukung MariaDB versi 10.3.13 sebagai sumber.
SysAdmin Non-akses ke sumber Microsoft SQL Server yang dikelola sendiri	<p>AWS DMS sekarang mendukung akses oleh SysAdmin non-pengguna ke titik akhir sumber SQL Server on-premise dan EC2 yang dihosting.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Fitur ini sekarang ada dalam versi Beta. Jika Anda ingin mencobanya, hubungi AWS dukungan untuk informasi lebih lanjut.</p> </div>
Tabel tugas CDC dan sumber Oracle dibuat menggunakan CREATE TABLE AS	AWS DMS sekarang mendukung tugas full-load dan CDC dan CDC saja yang berjalan melawan tabel sumber Oracle yang dibuat menggunakan pernyataan tersebut. CREATE TABLE AS

Masalah yang diselesaikan di AWS DMS 3.4.0 meliputi yang berikut:

- Memperbaiki penilaian tugas pra-migrasi. Untuk informasi selengkapnya, lihat [Mengaktifkan dan bekerja dengan penilaian perdana untuk tugas](#).
- Perbaiki validasi data untuk tipe data float, real, dan ganda.
- Perbaiki Amazon Redshift sebagai target dengan penanganan kesalahan ini dengan lebih baik: “Kunci yang ditentukan tidak ada.”

- Mendukung pengaturan tugas pemuatan CDC multithreaded, termasuk `ParallelApplyThreads`, `ParallelApplyBufferSize`, `ParallelApplyQueuesPerThread` untuk Amazon OpenSearch Service (OpenSearch Service) sebagai target.
- Peningkatan OpenSearch Layanan sebagai target dengan mendukung penggunaan kunci primer komposit.
- Memperbaiki masalah di mana koneksi uji gagal saat menggunakan PostgreSQL sebagai sumber dan kata sandi memiliki karakter khusus di dalamnya.
- Memperbaiki masalah dengan menggunakan SQL Server sebagai sumber ketika beberapa kolom VARCHAR terpotong.
- Memperbaiki masalah saat AWS DMS tidak menutup transaksi terbuka saat menggunakan Amazon RDS SQL Server sebagai sumber. Hal ini dapat mengakibatkan kehilangan data jika parameter interval pemungutan suara diatur dengan benar. Untuk informasi selengkapnya tentang cara mengatur rekomendasi nilai interval pemungutan suara, lihat [Menggunakan database Microsoft SQL Server sebagai sumber AWS DMS](#).
- Memperbaiki masalah untuk Oracle Standby sebagai sumber di mana tugas CDC akan berhenti tiba-tiba saat menggunakan Pembaca Biner.
- Memperbaiki masalah untuk IBM DB2 untuk LUW di mana tugas gagal dengan pesan “Numerik literal 0 tidak valid karena nilai di luar jangkauan.”
- Memperbaiki masalah untuk migrasi PostgreSQL ke PostgreSQL ketika kolom baru ditambahkan pada sumber PostgreSQL dan kolom dibuat dengan tipe data yang berbeda dari tipe data yang kolom awalnya dibuat pada sumber tersebut.
- Memperbaiki masalah dengan sumber MySQL di mana tugas migrasi tiba-tiba berhenti saat tidak dapat mengambil binlog.
- Memperbaiki masalah terkait target Oracle saat `BatchApply` sedang digunakan.
- Memperbaiki masalah untuk MySQL dan MariaDB saat memigrasi tipe data. TIME
- Memperbaiki masalah untuk sumber IBM DB2 LUW di mana migrasi tabel dengan LOB gagal ketika tabel tidak memiliki kunci primer atau kunci unik.

AWS Catatan rilis Database Migration Service 3.3.4

Masalah yang diselesaikan di AWS DMS 3.3.4 meliputi yang berikut:

- Memperbaiki masalah saat transaksi dibatalkan atau diduplikasi saat menggunakan PostgreSQL sebagai sumber.

- Meningkatkan support menggunakan tanda dolar (\$) dalam nama skema.
- Memperbaiki masalah di mana instans replikasi tidak menutup transaksi terbuka saat menggunakan RDS SQL Server sebagai sumber.
- Memperbaiki masalah di mana koneksi uji gagal saat menggunakan PostgreSQL sebagai sumber dan kata sandi memiliki karakter khusus di dalamnya.
- Perbaiki Amazon Redshift sebagai target dengan penanganan kesalahan ini dengan lebih baik: "Kunci yang ditentukan tidak ada."
- Peningkatan support validasi data pada berbagai tipe data dan pengaturan karakter.
- Memperbaiki masalah di mana tugas replikasi mencoba membuat tabel yang ada untuk Oracle sebagai target.
- Memperbaiki masalah di mana tugas replikasi tidak menambahkan "D" untuk kolom "OP" dalam syarat-syarat tertentu untuk Amazon S3 sebagai target.

AWS Catatan rilis Database Migration Service 3.3.3

Tabel berikut menunjukkan fitur baru dan memperkenalkan perangkat tambahan di AWS DMS versi 3.3.3.

Fitur baru atau perangkat tambahan	Deskripsi
Versi PostgreSQL baru	PostgreSQL versi 12 sekarang didukung sebagai sumber dan target.
Support untuk kunci primer komposit dengan Amazon OpenSearch Service sebagai target	Pada AWS DMS 3.3.3, penggunaan kunci primer komposit didukung oleh target OpenSearch Layanan.
Support untuk tipe data Oracle yang berkelanjutan	Tipe data Oracle yang berkelanjutan untuk sumber dan target Oracle sekarang menerima support.
Peningkatan jumlah sumber AWS DMS daya per akun	Batas jumlah AWS DMS sumber daya yang dapat Anda buat telah meningkat. Untuk informasi selengkapnya, lihat Kuota untuk AWS Database Migration Service .

Masalah yang diselesaikan di AWS DMS 3.3.3 meliputi yang berikut:

- Memperbaiki masalah di mana tugas gagal menggunakan pernyataan pembaruan khusus dengan Parallel Apply di Amazon Kinesis.
- Memperbaiki masalah di mana tugas gagal pada pernyataan ALTER TABLE dengan Amazon S3 sebagai target.
- Memperbaiki masalah di mana nilai-nilai pada kolom poligon terpotong saat menggunakan Microsoft SQL Server sebagai sumber.
- Memperbaiki masalah pada konverter Unicode JA16SJITILDE dan JA16EUCTILDE saat menggunakan Oracle sebagai sumber.
- Memperbaiki masalah di mana kolom MEDIUMTEXT dan LONGTEXT gagal melakukan migrasi dari format MySQL ke S3 nilai (CSV) yang dipisahkan dengan koma.
- Memperbaiki masalah di mana kolom boolean diubah menjadi tipe yang salah dengan output Apache Parquet.
- Memperbaiki masalah dengan kolom varchar yang berkelanjutan di Oracle.
- Memperbaiki masalah di mana tugas validasi data gagal karena kombinasi stempel waktu tertentu.
- Memperbaiki masalah dengan replikasi Sybase data definition language (DDL).
- Memperbaiki masalah yang melibatkan sumber Klaster Aplikasi Oracle Real (RAC) berbenturan dengan Pembaca Biner Oracle.
- Memperbaiki masalah dengan validasi untuk target Oracle dalam hal nama skema.
- Memperbaiki masalah dengan validasi IBM Db2 versi 9.7 dan 10.
- Memperbaiki masalah untuk tugas yang bukan pemfilteran stopword dua kali dengan mengaktifkan StopTaskCachedChangesApplied dan StopTaskCachedChangesNotApplied.

Riwayat dokumen

Tabel berikut menjelaskan perubahan penting pada dokumentasi panduan pengguna Database Migration Service AWS setelah Januari 2018.

Untuk pemberitahuan tentang pembaruan dokumentasi ini, Anda dapat berlangganan umpan RSS. Untuk detail selengkapnya tentang rilis AWS DMS versi, lihat [AWS Catatan rilis DMS](#).

Perubahan	Deskripsi	Tanggal
AWS DMS menambahkan dukungan untuk RDS IBM DB2 sebagai target	AWS DMS sekarang mendukung pengguna Amazon RDS IBM DB2 sebagai target.	Desember 4, 2023
AWS DMS menambahkan dukungan untuk Timestream sebagai target.	AWS DMS sekarang mendukung Timestream sebagai target.	17 November 2023
AWS DMS menambahkan dukungan untuk validasi data target Redshift	AWS DMS sekarang mendukung validasi data dalam target Redshift.	14 November 2023
AWS DMS menambahkan dukungan untuk empat tipe endpoint baru	AWS DMS sekarang mendukung penggunaan Microsoft Azure Database untuk PostgreSQL, Microsoft Azure Database untuk MySQL, OCI MySQL Heatwave, dan Google Cloud untuk PostgreSQL sebagai sumber.	26 Oktober 2023
AWS DMS menambahkan dukungan untuk peran terkait AWS layanan baru	AWS DMS sekarang mendukung peran AWS terkait layanan AWS Service RoleForDMSServerless yang memungkinkan	22 Mei 2023

	<p>AWS DMS untuk membuat dan mengelola sumber daya atas nama Anda, seperti menerbitkan titik data metrik ke Amazon. CloudWatch</p>	
<p>AWS DMSmenambahkan dukungan untuk kebijakan AWS terkelola baru</p>	<p>AWS DMSsekarang mendukung kebijakan AWS terkelola yang memungkinkan penerbitan log replikasi tanpa server ke Log. CloudWatch</p>	<p>22 Mei 2023</p>
<p>AWS DMSmenambahkan dukungan untuk kebijakan AWS terkelola baru</p>	<p>AWS DMSsekarang mendukung kebijakan AWS terkelola yang memungkinkan penerbitan titik data metrik ke Amazon CloudWatch. Juga, AWS DMS mulai melacak perubahan untuk kebijakan yang AWS dikelola.</p>	<p>Maret 6, 2023</p>
<p>Support sumber VPC dan titik akhir target</p>	<p>AWS DMSsekarang mendukung titik akhir virtual private cloud (VPC) sebagai sumber dan target. AWS DMSsekarang dapat terhubung ke AWS layanan apa pun dengan titik akhir VPC ketika rute yang ditentukan secara eksplisit ke layanan ditentukan dalam VPC mereka. AWS DMS</p>	<p>30 Juni 2022</p>

[Support SQL Server membaca replika sebagai sumber](#)

AWS DMS sekarang mendukung SQL Server membaca replika sebagai sumber. Dengan menggunakan AWS DMS, Anda sekarang dapat melakukan migrasi langsung dari replika baca SQL Server ke target yang AWS DMS didukung.

30 Juni 2022

[Support IBM Db2 z/OS database sebagai sumber untuk full load saja](#)

AWS DMS sekarang mendukung database IBM Db2 z/OS sebagai sumber. Dengan menggunakan AWS DMS, Anda sekarang dapat melakukan migrasi langsung dari mainframe Db2 ke target yang didukung. AWS DMS

30 Juni 2022

[Mendukung EventBridge acara DMS](#)

AWS DMS mendukung pengelolaan langganan acara yang digunakan EventBridge untuk acara DMS.

30 Juni 2022

[Support Babelfish sebagai target](#)

AWS DMS sekarang mendukung Babelfish sebagai target. Dengan menggunakan AWS DMS, Anda sekarang dapat memigrasikan data langsung dari sumber apa pun yang AWS DMS didukung ke Babelfish, dengan waktu henti minimal.

30 Juni 2022

Support Aurora Serverless v2 sebagai target	AWS DMS sekarang mendukung Aurora Serverless v2 sebagai target. Dengan menggunakan AWS DMS, Anda sekarang dapat melakukan migrasi langsung ke Aurora Tanpa Server v2.	30 Juni 2022
Memulai Tutorial	Pembaruan Tutorial Awal untuk AWS DMS Database Migration Service. Tutorial menggunakan basis data MySQL sebagai sumber dan basis data PostgreSQL sebagai target.	20 Mei 2021
Support untuk Amazon Neptune sebagai target	Dukungan tambahan untuk Amazon Neptune sebagai target migrasi data.	1 Juni 2020
Support untuk Apache Kafka sebagai target	Dukungan tambahan untuk Apache Kafka sebagai target untuk migrasi data.	20 Maret 2020
Konten keamanan yang diperbarui	Konten keamanan yang diperbarui dan distandarisasi sebagai respons terhadap permintaan pelanggan.	20 Desember 2019
Migrasi dengan AWS Snowball Edge	Dukungan tambahan untuk menggunakan Edge AWS Snowball untuk migrasi database besar.	24 Januari 2019

<u>Support untuk Amazon DocumentDB (dengan kompatibilitas MongoDB) sebagai target</u>	Dukungan tambahan untuk Amazon DocumentDB (dengan kompatibilitas MongoDB) sebagai target migrasi data.	9 Januari 2019
<u>Dukungan untuk Amazon OpenSearch Service dan Amazon Kinesis Data Streams sebagai target</u>	Menambahkan dukungan untuk OpenSearch Service dan Kinesis Data Streams sebagai target migrasi data.	15 November 2018
<u>Dukungan awal asli CDC</u>	Dukungan tambahan untuk titik awal asli saat menggunakan perubahan pengambilan data (CDC).	28 Juni 2018
<u>Dukungan Db2 LUW</u>	Dukungan tambahan untuk IBM Db2 LUW sebagai sumber migrasi data.	26 April 2018
<u>SQL Server sebagai dukungan target</u>	Dukungan tambahan untuk Amazon RDS for Microsoft SQL Server sebagai sumber.	6 Februari 2018

AWSGlosarium

Untuk AWS terminologi terbaru, lihat [AWSglosarium di Referensi](#). Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.